UNIVERSIDAD AUTONOMA DE MADRID

ESCUELA POLITECNICA SUPERIOR

Grado en Ingeniería de Tecnologías y Servicios de Telecomunicación

TRABAJO FIN DE GRADO

RECONOCIMIENTO AUTOMÁTICO DE ACENTO MEDIANTE I-VECTORS DE FRECUENCIAS FORMANTES EN UNIDADES LINGÜÍSTICAS

Álvaro Cañas Arranz
Tutor: Javier Franco Pedroso
Ponente: Joaquín González Rodríguez

Junio 2018

RECONOCIMIENTO AUTOMÁTICO DE ACENTO MEDIANTE I-VECTORS DE FRECUENCIAS FORMANTES EN UNIDADES LINGÜÍSTICAS

AUTOR: Álvaro Cañas Arranz
TUTOR: Javier Franco Pedroso

Escuela Politécnica Superior
Universidad Autónoma de Madrid
Junio de 2018

Resumen

En ocasiones, los sistemas de reconocimiento de voz pueden presentar problemas si el acento de la señal de voz que les llega no coincide con el acento de la señal de voz con el que fueron entrenados. Si se consiguiese elaborar un sistema automático de detección de acentos y elaborar reconocedores de voz a partir de ellos que tuviesen en cuenta las variaciones fonéticas según el acento de entrada, se reduciría en gran medida el problema enunciado anteriormente.

Estos nuevos sistemas de reconocimiento de voz, serían, por tanto mucho menos restrictivos a las señales de voz que les llegan y debiendo ofrecer un buen rendimiento ante las variaciones del acento de la señal de entrada.

Por otra parte es interesante llevarnos este concepto al campo forense, donde a veces se realizan tareas de caracterización de locutores (*speaker profiling*) para ayudar a la resolución del caso extrayendo ciertas características de un locutor tales como: edad, sexo, origen, etc, a partir de una grabación de voz. En el caso que nos incumbe para '*speaker profiling*', nos ayudaría a extraer información relevante sobre el origen geográfico del locutor a través de su acento.

El objetivo de este Trabajo Fin de Grado es aplicar al reconocimiento automático de acento una aproximación que ha sido usada con éxito en reconocimiento de locutor y que permite analizar las contribuciones individuales de distintas unidades lingüísticas en el proceso de reconocimiento, lo que podría aportar información útil (como por ejemplo, que unas unidades son más útiles para reconocer un acento que otro).
Para llevar a cabo dicho objetivo, nos centraremos a lo largo de trabajo en diseñar un sistema basado en unidades lingüísticas y compararlo con un sistema de referencia.

Abstract

Sometimes, systems based on voice recognition can have some problems if the accent of the entry voice signal doesn't match with the accent of the voice signal that trained the system. If we could elaborate an automatic accent recognition system and elaborate voice recognition systems based on previous one that kept in mind the phonetic variations depending on the entry accent, we could reduce the previous problem mentioned. Thus, these recognizers would be more complete because it considers more characteristics about speakers.

Those new voice recognition systems could be much more flexibles with the entry voice signals and could offer high efficiency despite accent variations of the entry signal.

On the other hand this concept is interesting in forensic task, where sometimes it is necessary to do speaker characterization tasks (speaker profiling) that help to solve the case extracting speaker characteristics as: age, sex, origin, etc. from unknown voice recording. In our task related in 'speaker profiling' it could help us to extract important information about the geographic origin of the speaker thanks to his accent.

In this TFG the main objective is looking for evaluate, observe and correct when it is necessary the efficiency of different systems based on i-vectors (and therefore based on formant frequencies in linguistic units) used previously in automatic speaker recognition task, and in this paper, used in automatic accent recognition task.

Palabras clave

I-vectors, frecuencias formantes, unidad lingüística, sistema automático, reconocimiento de voz/acento.

Keywords

I-vectors, formant frequencies, linguistic units, automatic system, voice/accent recognition.

Agradecimientos

En primer lugar agradecer a mi tutor Javier por haberme propuesto este gran TFG, por su paciencia, confianza y sobretodo ayuda prestada. Muchas gracias.

También agradecer al resto de profesores que he tenido a lo largo de la carrera y que me han hecho adquirir grandes conocimientos a lo largo de estos 4 años.

También agradecer a todos mis compañeros de clase, tanto los que están como los que tuvieron que irse por estos increíbles años donde hemos vivido de todo pero sobretodo buenos momentos. Tenemos amistad para rato.

Tampoco pueden faltar mis amigos de toda la vida; con quienes he contado siempre para contarles mis penas y alegrías durante estos 4 años y de las que nos hemos acabado riendo siempre. Gracias por todo y por siempre.

Por último agradecer a los de casa, a mi familia, sin ellos no estaría ahora donde estoy. Muchas gracias.

INDICE DE CONTENIDOS

INDICE DE FIGURAS

INDICE DE TABLAS

1 Introducción

1.1 Motivación

El reconocimiento de acento es un asunto de gran interés principalmente para su uso en sistemas automáticos de reconocimiento de voz [1]. Pueden mejorar las prestaciones de estos y aumentar su calidad debido a que tienen en cuenta un factor de variabilidad adicional: el acento de la señal de entrada.
Esto podría llevarse a cabo teniendo reconocedores de voz en un idioma entrenado con distintos acentos, y en función del que se detectase se enviaría a uno u otro sistema.

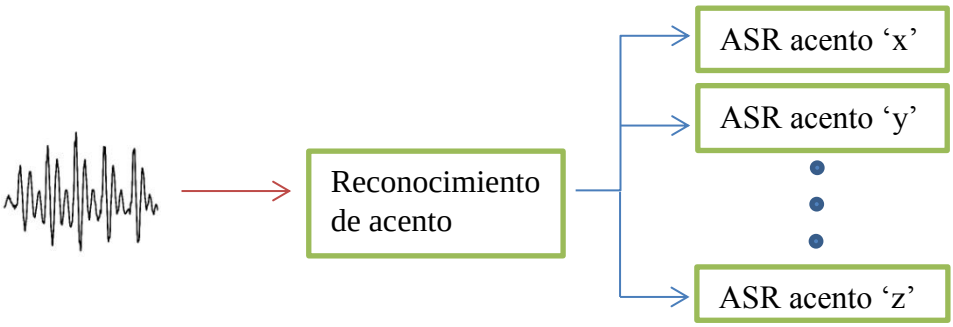

Figura 1.1: Diagrama de reconocedor de voz en un idioma entrenado con diferentes acentos.

El reconocimiento de acento supone otra vuelta de tuerca a la extracción de información del locutor, además de algunas características básicas como pueden ser por ejemplo la edad, sexo etc., este método nos puede ofrecer información relevante sobre el origen geográfico del locutor y su lengua materna.

A lo largo de este trabajo, nos referiremos al acento como la lengua materna del locutor al cual pertenecen algunas grabaciones que ha hecho él mismo en inglés.

El acento es por lo tanto una característica común a un grupo de personas de la misma zona geográfica, y que puede ser de gran interés clasificar e identificar (por ejemplo en el campo forense, o en los llamados "*call centers*", donde se detecte que alguien hablando en inglés tiene acento de España, por ejemplo, y le pasen con un operador que hable español). A lo que identificación de acento se refiere, realmente se podrá hacer en mayor o menor medida, dependiendo del nivel de inglés que tengan los distintos locutores de cada zona con el que han participado en las grabaciones. En otras palabras, no tiene la misma dificultad identificar el acento de una grabación en inglés de un locutor con un nivel bajo de inglés, que uno que tenga un nivel más alto (en ocasiones que no se identifique ni siquiera su lengua materna), en cuyo caso la probabilidad de error en la identificación de su acento será mucho mayor.

1.2 Objetivos

En este trabajo se crea un sistema que pretende actuar como un reconocedor de acentos a partir de ficheros de grabaciones pertenecientes a distintos locutores y acentos. Se busca que nuestro sistema sea capaz de averiguar el acento de los distintos ficheros de grabaciones a partir de un método que nos sirva como base o referencia y más tarde con una aproximación diferente a partir de unidades lingüísticas y formantes. Los objetivos específicos son:

1) Diseñar un entorno y protocolo experimental para poder evaluar distintas implementaciones de sistemas automáticos de reconocimiento de acento.

2) Implementar un sistema de referencia basado en técnicas en el estado del arte, y medir su rendimiento en el entorno experimental diseñado

3) Implementar un sistema basado en la aproximación propuesta, que hace uso de información lingüística para llevar a cabo el proceso de reconocimiento de acento. Medir su rendimiento en el entorno experimental diseñado y comparar con el sistema de referencia.

4) Evaluar el efecto sobre el rendimiento de distintas técnicas de compensación de canal o normalización, tanto para el sistema de referencia como para la aproximación propuesta

1.3 Organización de la memoria

La memoria consta de los siguientes capítulos:

- **Capítulo 1: introducción**. En este capítulo se hace una presentación del problema a abordar durante el trabajo, se marcan los objetivos, y se encuentra la organización de esta memoria.

- **Capítulo 2: estado del arte.** En este capítulo se explican las técnicas y conceptos teóricos más importantes empleadas hasta ahora en el campo de reconocimiento de acento y que han servido como base en el desarrollo de este trabajo.

- **Capítulo 3: diseño.** En este capítulo se presentan en detalle los sistemas empleados en este trabajo así como su funcionamiento.

- **Capítulo 4: desarrollo.** En este capítulo se especifica la base de datos y la partición de dichos datos empleados en el trabajo para hacer funcionar los sistemas propuestos

- **Capítulo 5: integración, pruebas y resultados.** En este capítulo se describen los resultados obtenidos así como la interpretación de los mismos para los sistemas propuestos y las mejoras empleadas.

- **Capítulo 6: conclusiones y trabajo futuro.** En este capítulo final se ponen de manifiesto las conclusiones más significativas derivadas del trabajo así como aspectos que podrían mejorar en un futuro los resultados del propio trabajo.

2 Estado del arte

2.1 Introducción

El reconocimiento de acentos es de especial importancia en el campo de reconocimiento de voz. Sin él, el rendimiento para hablantes no nativos experimenta una caída significativa [1][6][7], por lo que en los últimos años, se viene utilizando y complementando a reconocedores de voz.
También es de gran importancia en el ámbito forense, inteligencia, seguridad y control de fronteras, debido a que el acento y/o el idioma de personas bajo estudio o sospechosas podrían ser muy variados y teniendo en cuenta el reconocimiento de acento podría facilitar la resolución de diferentes casos [2].

Hay que recordar que nos referiremos a acento como la lengua materna de los locutores correspondientes a las diferentes grabaciones en inglés con las que trabajamos.

El acento está influenciado por nuestra lengua materna, por ejemplo un locutor bilingüe de inglés-español probablemente tendrá un acento inglés al hablar en español debido a que aprendió español más tarde en su vida. Los locutores no nativos introducen variaciones en la forma de pronunciar palabras, o variaciones en la estructura gramatical (aunque esto no se detectaría con un sistema acústico como los que usamos aquí, que sólo se fija en los sonidos, pero hay otros sistemas que sí podrían hacerlo porque analizan secuencias de palabras) en su segunda lengua, pero curiosamente estas variaciones no son aleatorias para los hablantes de una lengua, si no que de forma general, suelen ser comunes, y que ayudan a la detección de su lengua materna o acento [2].

2.2. Aproximaciones fonotácticas y acústicas.

En la actualidad se utilizan principalmente dos aproximaciones para abordar el problema de reconocimiento de acento: fonotácticas y acústicas.

La aproximación fonotáctica utiliza las frecuencias de aparición de secuencias de fonemas para reconocer acentos o, como también se ha usado en el pasado, reconocimiento de idioma [1].
Los sistemas fonotácticos funcionan de la siguiente forma:

- Usan un reconocedor de fonemas de un idioma que no tiene por qué ser el que se hable en la grabación.
- Luego, a partir esas decodificaciones fonéticas, modelan secuencias de fonemas para un idioma dado.

Ejemplo: con un reconocedor fonético de inglés hacemos la "decodificación fonética" de grabaciones en distintos idiomas: español, francés y portugués. Para cada idioma, analizamos cuáles son las secuencias de fonemas (del inglés) que aparecen con más frecuencia. Si en una grabación de test aparecen muchas secuencias de fonemas que son frecuentes en español (o más que en los otros idiomas), lo identificamos como español.

El ejemplo anterior sería un claro ejemplo de uso del método fonotáctico PRLM (*phone recognizer followed by langage models)*. También se suele emplear otro método parecido [1], PRLM paralelo (PPRLM, *parallel PRLM*) que es prácticamente lo mismo pero con varios decodificadores fonéticos en paralelo (tendríamos tantos modelos de un idioma como decodificadores fonéticos), cuyas salidas se combinarían.

Por otro lado, las aproximaciones acústicas, utilizan información tomada directamente de las características espectrales de las señales de audio, principalmente se utilizan los MFCCs (*mel-frequency cepstral coefficient)*, frecuentemente combinados con coeficientes derivados SDC (*shifted delta cepstra).* Estos sistemas tienen la ventaja de no requerir decodificadores fonéticos de un idioma específico, respecto a las aproximaciones fonotácticas.

Puesto que las características fonotácticas y las características acústicas producen resultados complementarios, actualmente se suelen combinar ambas técnicas a través de la fusión de los resultados de cada una.

En este trabajo nos centraremos únicamente en las aproximaciones acústicas, más en concreto en los *i-vectors*, que se tratan de una representación vectorial de dimensión fija de una secuencia de voz (grabación) de duración variable.
Los *i-vectors* se basan en técnicas de modelado usadas anteriormente como son las aproximaciones GMM y GMM-UBM.

2.3. Aproximaciones acústicas: GMM (Gaussian Mixture Models)

Para la gran mayoría de las tareas de reconocimiento, necesitamos modelar
la distribución de las características de los vectores con los que vamos a trabajar. Una técnica muy habitual son los modelos de mezclas de gaussianas (GMM, *Gaussian Mixture Models).*

Un GMM no es más que una suma ponderada de funciones de densidad de probabilidad Gaussianas:

$$p(\vec{x}|\lambda_s) = \sum_{i=1}^{M} p_i \, b_i(\vec{x})$$

$$\lambda_s = (p_i, \vec{\mu}_i, \sum_i)$$

Donde,

$p_i \equiv$ Peso (ponderación) de la componente i.
$\vec{\mu}_i \equiv$ vector de medias de la componente i.
$\sum_i \equiv$ matriz de covarianzas de la componente i.

$$b_i(\vec{x}) = \frac{1}{(2\pi)^{D/2} \, |\sum_i|^{1/2}} \exp(-\frac{1}{2}(\vec{x} - \vec{\mu}_i)' \, \sum_i{}^{-1} \, (\vec{x} - \vec{\mu}_i))$$

El uso de GMMs requiere dos cosas:
-Estimar los parámetros de un GMM dado un conjunto de vectores de características. Para obtener el modelo de locutor/idioma/acento.

-Calcular la probabilidad de una secuencia de características dado un GMM. Para obtener la puntuación de una grabación (secuencia de características) dado un modelo de locutor/idioma/acento.

Si asumimos que existe independencia entre los vectores de características en una secuencia, podemos calcular la probabilidad como:

$$p(\vec{x}, \dots, \vec{x}_N|\lambda) = \prod_{n=1}^{N} p(\vec{x}_N|\lambda)$$

Comúnmente escrito en notación logarítmica:

$$log\ \ p(\vec{x}, \dots, \vec{x}_N|\lambda) = \sum_{n=1}^{N} log\, p(\vec{x}_N|\lambda) = \sum_{n=1}^{N} log\ \left(\sum_{i=1}^{M} p_i b_i(\vec{x})\right)$$

Los parámetros de los GMMs se estiman maximizando la probabilidad de un conjunto de vectores de entrenamiento:

$$\lambda^* = \arg max_\lambda \sum_{n=1}^{N} log\ \ p(\vec{x}_N|\lambda)$$

Estableciendo las derivadas con respecto a los parámetros del modelo a cero y resolviendo:

$$p_i = \frac{1}{N} \sum_{n=1}^{N} Pr(i|\vec{x}_N)$$

$$\vec{\mu}_i = \frac{1}{n_i} \sum_{n=1}^{N} Pr(i|\vec{x}_N)\, \vec{x}_N$$

$$\sum_i = \frac{1}{n_i} \sum_{n=1}^{N} Pr(i|\vec{x}_N)\, \vec{x}_i \vec{x}_i{}' - \vec{\mu}_i \vec{\mu}_i{}'$$

Donde,

$$Pr(i|\vec{x}) = \frac{p_i b_i(\vec{x})}{\sum_{j=1}^{M} p_j b_j(\vec{x})}$$

$$n_i = \sum_{n=1}^{N} Pr(i|\vec{x}_t)$$

2.3.1. GMM-UBM *(Gaussian Mixture Models- Universal Background Model)*

El problema de entrenar directamente modelos GMM (como hemos visto en el apartado anterior) a partir de las características es que normalmente no tenemos muchos datos de la clase que queremos modelar (locutor/idioma/acento). Por eso es necesario entrenar un GMM "genérico" (UBM- *Universal Background Model*).
Este UBM posteriormente se adapta al locutor/idioma/acento que queremos modelar con los pocos datos de entrenamiento que tengamos mediante la adaptación MAP.

2.3.1.1 Adaptación MAP (*Maximum A-Posteriori*)

El modelo '*target'* se suele entrenar adaptándose desde el modelo '*background'* a través de la adaptación MAP [5].
Para llevar a cabo el entrenamiento con MAP, es necesario alinear los vectores de entrenamiento del locutor '*target'* con el UBM, acumular los estadísticos suficientes y obtener los parámetros del modelo (medias y covarianzas) para los datos de entrenamiento dados. Finalmente los parámetros del modelo adaptado se obtienen ponderando las nuevas medias y covarianzas junto con los parámetros del UBM.

Hay que tener en cuenta que con MAP sólo se actualiza los parámetros que representan fenómenos acústicos vistos en los datos de entrenamiento del locutor/idioma/acento '*target'*.

El uso de la adaptación MAP proporciona dos grandes ventajas:
-Mantiene la correspondencia entre las componentes de los modelos 'target' y el UBM. Esto permite obtener las puntuaciones (*scores)* de forma más rápida usando sólo las componentes que acumulan mayor probabilidad (*top-M-scoring)* que se asumen iguales para ambos modelos.

-Proporciona una 'hipótesis alternativa' común a todos los locutores (el UBM) en el cálculo de la relación de verosimilitudes.

2.3.1.2 *Scoring* entre el modelo *target* y el UBM (*Universal Background Model*)

Los GMMs se utilizan para definir tanto el modelo '*targe*t' como el modelo '*background*'. Para el problema de reconocimiento de de acento, el modelo '*target'* sería un acento en

concreto, mientras que el modelo '*background*' serían grabaciones en muchos acentos (o todos los que estamos considerando en la prueba).

-El modelo '*target*' entrenado usa la voz inscrito.
-El modelo '*background'* entrenado utiliza la voz de varios locutores. A este modelo se le suele denominar UBM (*Universal Background Model).*

La puntuación final se obtiene mediante la relación de verosimilitudes en valor logarítmico (*log-likelihood ratio, LRR)* entre ambos modelos:

$$LRR = \Lambda = log\ p(X|target) - log\ p(X|\overline{target})$$

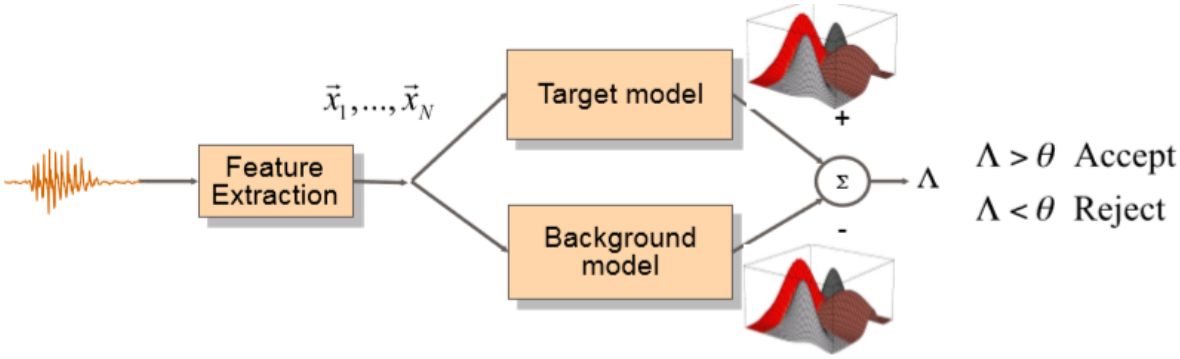

Figura 2.4.1: Explicación del umbral de aceptación/no aceptación de LRR a partir de los dos tipos de modelo (Fuente: Low-dimensional speech representation based on Factor Analysis and its applications- Najim Dehak and Stephen Shum)

2.3.2. GMM-SV *(GMM-supervector)*

En el proceso de adaptación MAP, es habitual adaptar únicamente los vectores de medias, a partir de los datos de entrenamiento del locutor, respecto a los del UBM, dejando inalterados el resto de parámetros (pesos y matrices de covarianza). Por lo tanto, la diferencia de un modelo de locutor respecto al UBM queda definida únicamente mediante dichos vectores de medias. Esto permite obtener una representación vectorial del modelo de locutor mediante la concatenación de dichos vectores de medias en un único vector de elevado número de dimensiones, denominado supervector. Aplicando el mismo proceso (adaptación MAP y formación de supervector) a las grabaciones de test, pueden representarse tanto la grabación del locutor 'target' como la grabación de test mediante vectores en un espacio de características común, lo que "simetriza" el proceso de obtención de puntuaciones. Además permite aplicar técnicas de compensación de variabilidad, modelado y clasificación asociadas a representaciones vectoriales.

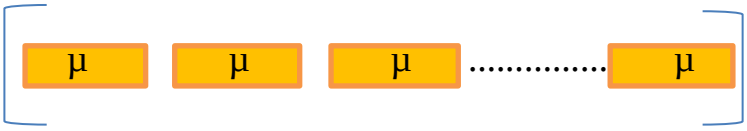

Figura 2.4.2: formación de los supervectores (concatenación de los vectores de medias).

2.3.3. i-vector

La técnica mediante GMS suele aportar buenos resultados y rendimiento principalmente en el campo de reconocimiento de locutor y por ello también se ha empleado en el campo de reconocimiento de acento [1].
Pero también diversos avances en este campo, han encontrado un método alternativo de modelar los Supervectores GMMs, que obtienen resultados con un rendimiento muy superior a la hora de identificar el acento de un locutor: el modelado de variabilidad total (*Total variability modeling).*
Un modelado de variabilidad total supone que el supervector de medias de un GMM, M, obtenido a partir de un conjunto de vectores de características de un locutor puede descomponerse como:

$M = m + Tw$ donde,

-m≡ supervector del UBM.
-T≡ matriz que define la proyección entre el espacio de alta dimensión de los supervectores y el subespacio de dimensión reducida de los *i-vectors.*
-w≡ factores (dimensiones del subespacio) que mejor representan la variación de M respecto a m.

El vector *w* es comúnmente llamado *i-vector* y tiene una distribución estándar normal *N*(0,I). La matriz T se estima a partir del análisis factorial (*factor analysis)* para representar las direcciones que presentan más variabilidad de unas locuciones a otras.

En el modelado de variabilidad total, los *i-vectors* son la representación de baja dimensionalidad de una grabación de audio que puede ser usada para tareas de clasificación y estimación.

2.3.3.1 Distancia Coseno (*Cosine Scoring*)

La distancia coseno se usa comúnmente para medir la similitud que existe entre dos *i-vectors*[2]. La distancia coseno 't' entre el *i-vector* de *test*, w_{test}, y el *i-vector* target de la clase (locutor/idioma/acento) 'a', w_{target}^{a}, se define como el producto escalar $<w_{test}, w_{target}^{a}>$ y se calcula como:

$$t = \frac{\widehat{w}_{test}^{T}\ \widehat{w}_{target}^{a}}{||\widehat{w}_{test}||\ ||\widehat{w}_{target}^{a}||} \qquad (1)$$

Donde, $||\widehat{w}||$ es la norma del *i-vector* y $\widehat{w}_{test}$ el *i-vector* de test.

Además $\widehat{w}_{target}^{a}$ es el *i-vector* promedio de todos los *i-vectors* de entrenamiento del acento 'a', es decir, el modelo del acento 'a':

$$\widehat{w}_{target}^{a} = \frac{1}{N_a}\sum_{i=1}^{N_a}\widehat{w}_i^a$$

Donde N_a es el número de grabaciones de entrenamiento del acento '*a'* y $\widehat{w}_i^a$ es el *i-vector* proyectado de la grabación de entrenamiento *i* del acento *a.*
La distancia coseno nos va a permitir en este trabajo obtener las puntuaciones entre los modelos de acento y los datos de test.

2.3.3.2 Técnicas de compensación y normalización

Las técnicas de compensación se utilizan principalmente para mejorar el rendimiento del sistema a través de la disminución de la dimensionalidad de los *i-vectors.*
En este trabajo utilizamos dos técnicas de compensación: LDA (*Linear Discriminant Analisys)* y WCCN (*Within-Class Covariance Normalization*), aunque justificaremos más adelante, porque descartamos finalmente la primera.

El principal objetivo de LDA es obtener una proyección del espacio de características (*i-vectors)* en la que aquellos correspondientes a distintos acentos sean más separables.
Se basa en encontrar una combinación lineal de rasgos que caracterizan o separan varias clases. La combinación resultante puede ser utilizada como un clasificador lineal, o, más comúnmente, para la reducción de dimensiones antes de la posterior clasificación.

WCCN [2] se utiliza con el propósito de compensar las posibles variaciones dentro de las clases en el espacio de variabilidad total (TV).
Se basa en el cálculo de una matriz que representa la covarianza intra-acento promediada entre los distintos acentos y se aplica sobre todo el conjunto de *i-vectors.*

Teniendo en cuenta las técnicas de compensación anteriores, si las aplicamos a los *i-vectors*, la distancia coseno del apartado anterior se calcularía haciendo uso de (1) con la diferencia de que:
$\widehat{w}_{test} = A^T\ w_{test}$ o $\widehat{w}_{test} = WCCN^T\ w_{test}$ (2)

Siendo A la matriz de proyección LDA y WCCN la matriz WCCN (en el caso que no empleemos una técnica u otra).

A parte de hacer uso de las dos técnicas de compensación mencionadas anteriormente, también hacemos uso de dos técnicas de normalización: *l-norm* y *whitening.*

l-norm consiste en una normalización del módulo que se aplica a todos los *i-vectors* de entrada y *whitening* en una normalización de media y varianza calculadas sobre un conjunto de *i-vectors* independiente (entrenamiento).

Además de incluir estas cuatro técnicas (2 de compensación y 2 de normalización), introducimos otra que consiste en normalizar todas las puntuaciones finales. A diferencia de las anteriores no afecta a los *i-vectors* (es independiente de los mismos), si no que afecta únicamente a los *scores* finales una vez llevado a cabo la distancia coseno.

Estas cuatro técnicas de compensación se describen más en detalle y acompañado de resultados en el capítulo 5.3.

2.4. Trabajos relacionados

2.4.1 Comparativa GMS, GPPS y *I-vectors*

En [1] se comparan tres métodos (GMS, GPPS, I-vector) junto con tres clasificadores en la tarea de reconocimiento de acento en un total de 6 acentos (ruso, vietnamita, cantonés, ingles de América, hindi y tailandés) en un total de 256 locutores procedentes de la base de datos NIST 2008 SRE. Las conclusiones fueron las siguientes:

-GMS (*Gaussian Mean Supervector)*: puesto que se basa en características de alta dimensionalidad (*high dimensionality*), presenta una gran dificultad en obtener sistemas robustos y un gran coste computacional. Presenta un peor resultado en la tarea de identificación de acento que los otros dos métodos siguientes [1].

-GPPS (*Gaussian Posterior Probability Supervector*): En lugar de trabajar únicamente con las medias (supervectores), modela la distribución completa. Además lleva información complementaria a GMS y por ello el rendimiento (*accuracy*) aumenta considerablemente .GPPS obtiene mejores resultados usando el clasificador SRC (*Sparse Representation Classifier*) [1][3].

-I-vector: se basa en una representación compacta en forma vectorial de baja dimensionalidad de una grabación. Obtiene buenos resultados en la tarea de reconocimiento de acento con el clasificador SVM (*Support Vector Machine*). Además se ha utilizado también en la tarea de reconocimiento de edad de locutores entre otras características [1][4].

En este trabajo nos centraremos únicamente en el uso de *i-vectors* para hacer el reconocimiento de la legua materna de los locutores.

2.4.2 ACCDIST (*Accent distance*)

ACCDIST se trata de una aproximación que usa características acústicas pero también información fonética. Se trata de una aproximación que busca mejorar las tasas de reconocimiento mediante la identificación de las características más útiles en una muestra de voz, dada cualquier colección de acentos [8].
ACCDIST aprovecha la información de un ASR y suele aportar resultados mejores en lo que a identificación de acento se refiere [5].
Existen dos sistemas que utilizan ACCDIST como aproximación: uno basado en la correlación (*Y-ACCDIST-Correlation*) y otro en SVM (*Y-ACCDIST-SVM*).

Y-ACCDIST-Correlation se basa en técnicas de correlación a través de distancias euclídeas entre fonemas vocálicos y se introducen en una matriz en forma de puntuaciones. Para cada clase de acento en la base de datos, se toma la matriz de cada hablante que pertenece al grupo y se calcula una matriz ACCDIST promedio para representar ese acento.
La correlación se calcula entre la matriz del hablante desconocida y cada una de las matrices de acentos representativas. La etiqueta de acento del hablante desconocido está determinada por la clase de acento con la que genera el mayor valor de correlación, lo que indica un mayor grado de similitud [8].

En Y-ACCDIST-SVM los locutores se procesan como se indicó anteriormente para modelar una matriz ACCDIST representativa para cada locutor. La diferencia con el sistema basado en correlación, sin embargo, radica en el proceso de clasificación.
Para cada acento, las matrices de los locutores que pertenecen a esa clase se modelan en un SVM (de la misma manera que el sistema GMM-SVM) y las matrices ACCDIST para todos los demás hablantes de todos los demás acentos se modelan para formar una configuración de 'uno contra el resto'.
Se forma un hiperplano óptimo para cada configuración entre la clase de acento y 'el resto'. Al clasificar una muestra de voz desconocida, se convierte en una matriz ACCDIST y posteriormente se incorpora a cada una de las SVM producidas para cada clase de acento. La etiqueta de clase de acento se decide en función del margen más claro formado entre el hablante desconocido y el hiperplano en cada uno de los SVM.

3 Diseño

3.1. Introducción

En este trabajo, la tarea para llevar a cabo el reconocimiento de acento se divide en dos partes. La primera es construir un sistema de referencia, que se pueda optimizar y conseguir el rendimiento más alto posible para que sirva como 'base' para comparar con la aproximación propuesta en este trabajo, que incorpora información lingüística en el proceso de reconocimiento.

3.2. Sistema de referencia

Con la creación del sistema de referencia, intentamos tener un sistema de partida basado en las técnicas mencionadas en el estado del arte con el que analizar los resultados mediante nuestra aproximación. Es una parte muy importante de este trabajo, puesto que sirve como base a todo lo que viene después y por ello ha de estar lo más optimizado y con el mejor rendimiento posible.

El sistema de referencia de este trabajo es un sistema creado a partir de aproximaciones acústicas, más en concreto en *i-vectors* de 600 dimensiones para cada grabación. A su vez estos *i-vectors* se han obtenido a partir de características basadas en MFCCs por lo que este sistema de referencia, es un sistema basado en características cepstrales, además de estar filtradas mediante *RASTA*, normalizadas y procesadas mediante CMN (*Cepstral Mean Normalized*) y *warping*, y consta de 19 coeficientes más deltas [9].

El proceso de reconocimiento consta de varios pasos que se explicarán más en detalle en el capítulo 5 acompañados de sus respectivos resultados:

-En primer lugar se crea el modelo de cada acento mediante el *i-vector* promedio de las grabaciones de entrenamiento para ese acento.
-Cómo veremos, con ese modelo de acentos, el sistema tendrá un error en la identificación de acentos excesivamente alto, por lo que habrá que introducir ciertas mejoras a través de técnicas de compensación y normalización.
- Una vez obtenido el modelo de acentos se obtienen las puntuaciones o *scores*: la puntuación de una grabación de test se obtiene mediante *cosine scoring* entre su *i-vector* y el *i-vector* medio del acento (modelo de acento).
- Para mejorar algo más los resultados del sistema, finalmente las puntuaciones se normalizan como se ha comentado brevemente en el apartado 2.3.3.2.

3.2.1 Obtención de las puntuaciones (*scores)*

Las puntuaciones marcan una similitud entre los datos de *test* y los modelos de acento.
La obtención de las puntuaciones o *scores* para el sistema de referencia incluye un proceso de proyección LDA tanto para los datos de *train,* usados en la creación de los modelos de acento, como los de *test* (capítulo 4.3).
Las matrices LDA se obtienen a partir de los datos de entrenamiento etiquetados. El principal objetivo de LDA es obtener una proyección del espacio de características (*i-vectors)* en la que aquellos correspondientes a distintos acentos sean más separables.
Los *scores* se obtienen mediante la comparación entre los datos de *train* que conforman los modelos de acento, y los datos de *test* mediante el cálculo de la distancia coseno usando la ecuación (1) y (2).

En este trabajo, hemos comprobado que el empleo de LDA no mejora el rendimiento de ninguno de los dos sistemas (de hecho los empeora un poco), posiblemente se deba a que únicamente trabajamos con 4 acentos (Capítulo 4.3), y al reducir las dimensiones, nos quedaríamos con bastante pocas, perdiendo mucha información.
Por ese motivo, para la obtención de las puntuaciones nos centraremos WCCN como técnica de compensación principal, y compararemos los resultados con los que nos proporcionaría LDA.

3.3. Sistema basado en unidades lingüísticas

Este sistema sigue el mismo esquema general que el sistema de referencia, pero a diferencia del anterior, en el que teníamos un único *i-vector* por grabación, ahora tenemos un *i-vector* para cada unidad lingüística, las cuales pueden aparecer (a veces múltiples veces) o no en las distintas grabaciones.
Esta aproximación que se ha utilizado antes en reconocimiento de locutor con buenos resultados [9], tiene la ventaja de proporcionar información interpretable (por ejemplo, se determina que una grabación es de un acento determinado porque cierta unidad lingüística se pronuncia de forma particular en dicho acento).

3.3.1 Información acústica modelada

Para el sistema basado en unidades lingüísticas que utilizamos en este trabajo, se utiliza un sistema de seguimiento de formantes para extraer las frecuencias formantes, que son las características acústicas que usamos en nuestra aproximación para este trabajo.
Las principales características del proceso de extracción de este sistema de seguimiento de formantes son las siguientes:

-Se calculan los valores de los tres primeros formantes cada 10ms.
-Se calculan sobre los formantes los coeficientes derivados (delta).

Estos coeficientes delta se añaden posteriormente a los vectores de características acústicas anteriores resultando tres formantes más sus deltas. Estos son los vectores de características acústicas de partida, a partir de los cuales se extraen los *i-vectors* para cada unidad lingüística.

3.3.2. Delimitación de unidades lingüísticas

En este trabajo, las etiquetas de fonema se generan de forma automática mediante un sistema de reconocimiento automático de voz que produce transcripciones que definen tanto el contenido fonético como el intervalo de tiempo de las regiones del habla en las que se puede segmentar el flujo de audio. En este trabajo, se utilizan las etiquetas de transcripción fonética producidas por el sistema de ASR (*automatic speaker recognition*) de SRIs Decipher [10].

3.3.3 Tipos de unidades lingüísticas

No todas las unidades lingüísticas van a tener las mismas características, por ello nos van a servir para marcar una serie de restricciones como son: tipo de sonido producido, frecuencia con las que aparecen en las grabaciones (no todas las unidades lingüísticas tienen porque aparecer en todas la grabaciones), longitud de la unidad, etc.

En este trabajo se utilizan dos tipos de unidades lingüísticas; los fonemas *y* los difonemas para modelar la información acústica (frecuencias formantes + deltas) en dichas unidades mediante *i-vectors*.

- Fonemas*:* son las unidades lingüísticas más cortas, y pueden aparecer en contextos lingüísticos diferentes. Normalmente suelen tener una alta frecuencia de aparición en las grabaciones, lo que permite elaborar sistemas más robustos.
  En este trabajo hemos utilizado 39 unidades lingüísticas (del inglés) de este tipo más dos "pausas llenas" (*'PUM', 'PUH')* [9].
  Estas unidades aparecen en la tabla 3.3.1.1 (para las vocales) y 3.3.1.2 (para las consonantes) representadas con dos letras (ej:*'AO'*).

- Difonemas*:* se pueden crear combinando cualquier pareja de fonemas. Se caracterizan por formar unidades lingüísticas más largas y tener una frecuencia de aparición menor, pero su contexto lingüístico es más reducido. En este trabajo se emplean los 98 difonemas más comunes.

Tabla 3.3.1.1: unidades lingüísticas (vocales) usadas junto con su transcripción al alfabeto fonético internacional (IPA), y junto a ejemplo de palabras en inglés (*Fuente: Linguistically-constrained formant-based i-vectors for automatic speaker recognition Javier Franco-Pedroso, Joaquín González-Rodríguez*)

VOCALES		
Unidad	**IPA**	**Ejemplo de palabra (inglés)**
AO	**ɔ**	**frost**
AE	**æ**	**fast**
AA	**a**	**father**
IY	**i**	**bee**
UW	**u**	**food**
EY	**eI**	**eight**
EH	**ɛ**	**red**
AY	**aI**	**ride**
IH	**I**	**win**
OW	**oʊ**	**show**
UH	**ʊ**	**should**
AW	**aʊ**	**now**
AH	**ʌ**	**but**
ER	**ɜ**	**heart**
AH	**ə**	**alone**
AX	**ə**	**discus**

Tabla 3.3.1.2: unidades lingüísticas (consonantes) usadas junto con su transcripción al alfabeto fonético internacional (IPA), y junto a ejemplo de palabras en inglés. (*Fuente: Linguistically-constrained formant-based i-vectors for automatic speaker recognition Javier Franco-Pedroso, Joaquín González-Rodríguez*)

CONSONANTES		
Unidad	**IPA**	**Ejemplo de palabra (en inglés)**
P	**p**	**pay**
B	**b**	**buy**
T	**t**	**take**
D	**d**	**day**
K	**k**	**key**
G	**g**	**go**
F	**f**	**for**
V	**v**	**very**
TH	**ɵ**	**thanks**
DH	**ð**	**that**
S	**s**	**say**
Z	**z**	**zoo**
SH	**ʃ**	**show**
HH	**h**	**house**
CH	**tʃ**	**chair**
JH	**dʒ**	**just**
Y	**j**	**yes**
W	**w**	**way**
L	**ɬ**	**late**
R	**r/ɹ**	**run**
DX	**ɾ**	**wetter**
M	**m**	**man**
N	**n**	**no**
NG	**ŋ**	**sing**

--- Stops
--- Fricativas
--- Africadas
--- Líquidas

3.4 Evaluación del rendimiento

En este trabajo se aborda el problema de reconocimiento de acento como un problema de detección o verificación en el que dado un acento 'objetivo', se analizan distintas grabaciones para comprobar si en ellas se habla con dicho acento. Por ello, se usan las medidas de rendimiento habituales para estos sitemas: curvas DET (*Detection Error Tradeoff)* y la tasa de igual error (*EER,Equal Error Rate)*[9] como medida de rendimiento 'global'. El EER indica el punto de trabajo del sistema en el que las tasas de falsa alarma y falsa aceptación son iguales. Finalmente, se evalúa el rendimiento global de cada aproximación (sistema de referencia y sistema propuesto) mediante el EER promedio sobre el conjunto de acentos de nuestro entorno experimental.

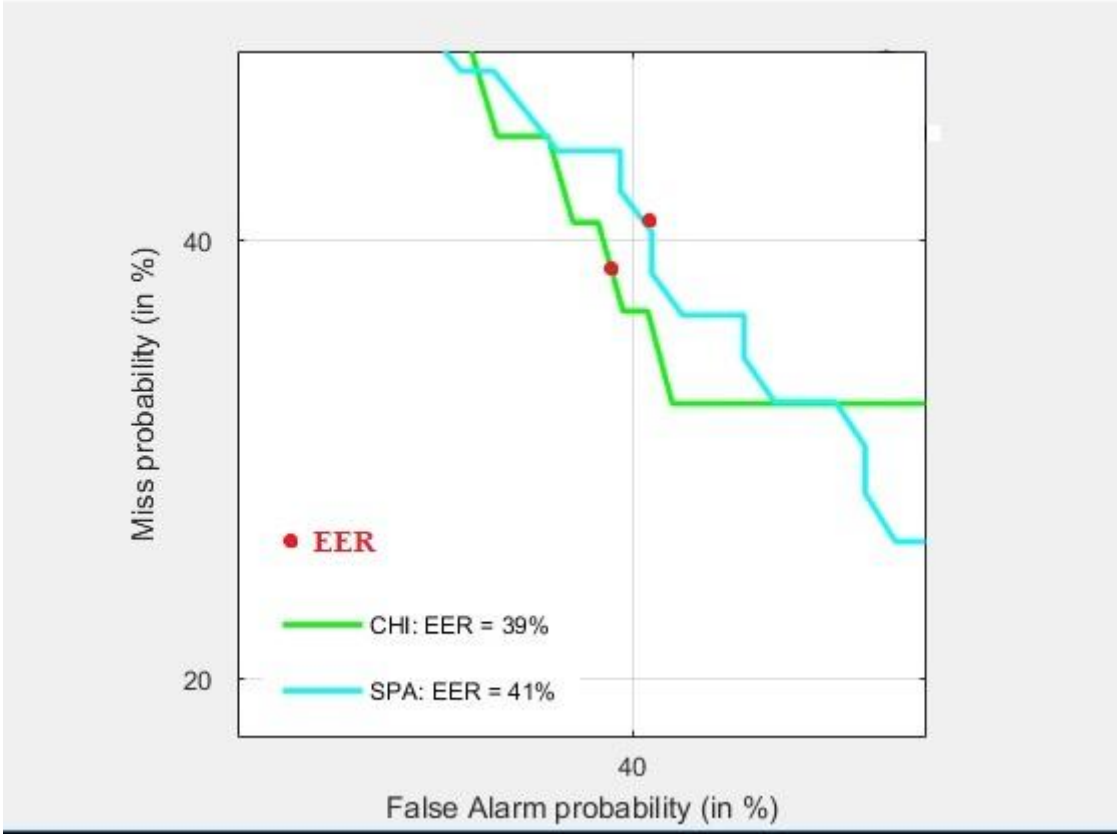

Figura 3.5: Ejemplo de EER en curva DET para acentos chino mandarín y español.

4 Desarrollo

4.1. Introducción

Para la creación de cualquier sistema automático de reconocimiento (bien sea voz, acento, o cualquier otra característica) es necesario definir la base de datos con la que se ha trabajado y hacer una separación de los datos con los que se va a entrenar y testear el sistema.
Es necesario definir cuáles van a ser las tres grandes particiones de datos con las que vamos a trabajar:
- entrenamiento: para obtener los modelos de cada acento (promedio de *i-vectors)* y otros parámetros del sistema (matrices WCCN y parámetros *whitening*).
- desarrollo: para realizar pruebas de reconocimiento que permitan ajustar la configuración del sistema (técnicas de compensación aplicadas, tipo de normalización de puntuaciones, etc.) de forma que se obtenga el mejor rendimiento posible.
- validación: para realizar pruebas de reconocimiento que validen la configuración del sistema sobre datos desconocidos.

En esta sección hablaremos de la base de datos empleada y como hemos hecho la separación de estos datos.

4.2. Base de datos

La base de datos empleada en este trabajo está formada por conjuntos de grabaciones utilizados en distintas evaluaciones de reconocimiento de locutor (*Speaker Recognition Evaluation*, SRE) organizadas por el *National Institute of Standards and Technology* (NIST) estadounidense.
Aunque se trata de bases de datos destinadas principalmente a tareas de reconocimiento de locutor, incorporan también otra información adicional como el idioma hablado en las grabaciones y la lengua materna de los locutores, lo que podemos aprovechar para desarrollar y evaluar sistemas de reconocimiento de acento.

Es necesario recordar y remarcar, que todas las grabaciones utilizadas son en inglés y que en este trabajo utilizamos la etiqueta de lengua materna como acento, es decir nos referiremos a los acentos como la lengua materna del locutor presente en cada grabación.
Por otro lado tenemos que tener muy en cuenta que este acento no tiene por qué parecerse mucho al de otro locutor de la misma zona geográfica (al de otro locutor con la misma lengua materna), puesto que el nivel de inglés del locutor que ha participado en las grabaciones puede ser mejor o peor.

Dentro de las grabaciones de NIST-SRE, se utilizan las correspondientes a los años 2004, 2005 y 2006. Esta base de datos NIST-SRE para los años mencionados, cuenta con un total de 14146 grabaciones en inglés, por locutores tanto masculinos como femeninos (también hay que tener en cuenta que un locutor puede haber grabado varias grabaciones, por lo que no existe el mismo número de grabaciones y de locutores), aunque para este trabajo únicamente hemos empleado locutores masculinos. Además consta de grabaciones de 10 segundos, 30 segundos y 300 segundos, con un total de 32 acentos. En este trabajo

únicamente nos centraremos en 4 de ellos: árabe, chino mandarín, ruso y español, que son los que están presentes en los tres conjuntos de datos (2004, 2005 y 2006).

4.3. Particiones

Como hemos dicho anteriormente, es necesario definir cuáles van a ser las tres grandes particiones de datos y con qué fin se van a usar para crear el protocolo experimental: datos de entrenamiento, datos de desarrollo y datos de validación con los que se va a entrenar y posteriormente evaluar respectivamente los sistemas diseñados.
Hay que tener en cuenta que para crear el sistema de referencia, sólo utilizaremos dos tipos de datos: los de *train* y los de desarrollo. Mientras que para el sistema basado en unidades lingüísticas utilizaremos tanto los de *train*, desarrollo y aparte los de validación para entrenar la fusión de dichas unidades. Por otro lado en el sistema de referencia todos los datos de *test* y *train* están presentes, mientras que para el sistema basado en unidades independientes, puede pasar que alguna unidad no se encuentre dentro de las grabaciones que se utilizan como *test*, *train* o desarrollo.
Habiendo quedado claro lo anterior, definimos las siguientes particiones:

- SRE-2004: datos de *train.*
- SRE-2005: datos de desarrollo.
- SRE-2006: datos de validación.

Datos de *train: corresponden a las grabaciones de SRE-2004 y se utilizan para entrenar los parámetros del sistema (principalmente el blanqueado y la normalización WCCN) y para construir los modelos de acento; ambos aspectos los veremos más en detalle en el capítulo 5.
En este trabajo tenemos 1951 grabaciones de *train*, de los cuales 548 pertenecen al acento árabe, 498 al chino mandarín, 346 al ruso y 559 al español.

***Datos de desarrollo**: corresponden a las grabaciones de SRE-2005 y se utilizan para evaluar el sistema para distintas configuraciones con el objetivo de encontrar aquella que dé lugar a un mejor rendimiento.
En este trabajo tenemos 143 grabaciones de desarrollo, de las cuales 41 pertenecen al acento árabe, 22 al chino mandarín, 33 al ruso y 47 al español.
Por lo tanto, el número de comparaciones que se establecen son:
41 comparaciones '*target*' y 143-41 '*non-target*' para el acento árabe.
22 comparaciones '*target*' y 143-22 '*non-target*' para el acento chino mandarín.
33 comparaciones '*target*' y 143-33 '*non-target*' para el acento ruso.
47 comparaciones '*target*' y 143-47 '*non-target*' para el acento español.

***Datos de validación**: corresponden a las grabaciones de SRE-2006 y se utilizan para verificar el rendimiento obtenido en las pruebas de desarrollo sobre un conjunto de datos desconocido.
En este trabajo tenemos 327 grabaciones de validación, de las cuales 19 pertenecen al acento árabe, 199 al chino mandarín, 73 al ruso y 36 al español.
19 comparaciones '*target*' y 327-19 '*non-target*' para el acento árabe.
199 comparaciones '*target*' y 327-199 '*non-target*' para el acento chino mandarín.
73 comparaciones '*target*' y 327-73 '*non-target*' para el acento ruso.
36 comparaciones '*target*' y 327-36 '*non-target*' para el acento español.

5 Integración pruebas y resultados

5.1 Introducción

En este capítulo vamos a centrarnos en cómo se ha realizado el trabajo en sí. Vamos a explicar desde donde partimos teniendo en cuenta todos los conceptos mencionados de los capítulos anteriores y hasta donde hemos llegado y con qué resultados.
Este capítulo corresponde por lo tanto a una parte más práctica donde nos apoyamos de toda la teoría de los capítulos anteriores para ir obteniendo resultados y mejorándolos.

5.2. Creación del sistema de referencia

Para la creación del sistema de referencia en primer lugar debemos tener los datos preparados según se ha especificado en el apartado 4.3. Estos datos vienen en forma de *i-vectors* directamente. Como también hemos comentado en el apartado 4.3, estos *i-vectors* los dividimos según sean datos de *train*, desarrollo o validación.
Los datos de *train* los vamos a utilizar para crear el modelo de acento.

5.2.1. Creación del modelo de acentos

Lo primero que hacemos es dividir cuantos datos hay de cada acento con el que trabajamos, es decir: 548 del árabe, 498 del chino mandarín, 346 del ruso y 559 del español y promediar todos los *i-vectors* correspondientes a cada acento para crear el *i-vector* modelo del árabe, del chino mandarín, del ruso y del español.
Eso bastaría y sería lo correcto si cada locutor de ese acento tuviera el mismo número de *i-vectors*, pero no tiene porqué ser así, y si algún locutor tiene muchos más *i-vectors* que el resto estará teniendo más "peso" en el modelo del acento. Para que el modelo de acento sea "más genérico" (todos los locutores tengan el mismo "peso") lo que hacemos es obtener primero el *i-vector* promedio para cada locutor individual (a partir de todos sus *i-vectors* de cada grabación en la que participa), y luego promediar los "*i-vectors* promedio" de todos los locutores para ese acento.
En resumen: una vez que tenemos los *i-vectors* correspondientes a cada locutor se hace la media de todos ellos para obtener un *i-vector* único para cada locutor, y posteriormente se vuelve a hacer la media de todos los que corresponden a un mismo acento para crear el modelo del propio acento.

5.2.2. *Scores* entre el modelo y los datos de *test*

Una vez que tenemos creado el modelo de cada acento y cargados todos los datos de *test*, vamos a obtener los *scores* correspondientes a todas las grabaciones respecto a ellos.
Para ello, utilizaremos la distancia coseno entre los datos de *test* y los modelos de acento creados.

Cada *i-vector* de test se compara con cada modelo mediante la distancia coseno, es decir se establecen 143x4 comparaciones (143 datos de *test* x 4 modelos) si se monta el sistema sobre los datos de desarrollo y 327x4 comparaciones si se hace sobre los datos de validación. De cada una de estas comparaciones obtenemos una puntuación o *score*, es decir, tenemos 4 *scores* diferentes para cada fichero de *test* correspondientes a cada modelo de acento que marca la similitud entre el propio modelo y el dato de *test.*
En primer lugar mostraremos los resultados del sistema de referencia (en curvas DET) así como la evolución de los mismos, sobre los datos de desarrollo y finalmente se mostrará el resultado final del sistema tanto sobre los datos de desarrollo como los de validación.

5.2.3. Resultados del sistema de referencia

Una vez que tenemos todos los *scores* disponibles, podemos hacer una primera evaluación final de los resultados. Como hemos dicho anteriormente en el capítulo 3.1, los resultados se van a evaluar según los EER medio correspondientes a las curvas DET, se busca que sea lo más bajo posibles. A continuación se muestran los resultados del sistema de referencia tras la aproximación mencionada anteriormente:

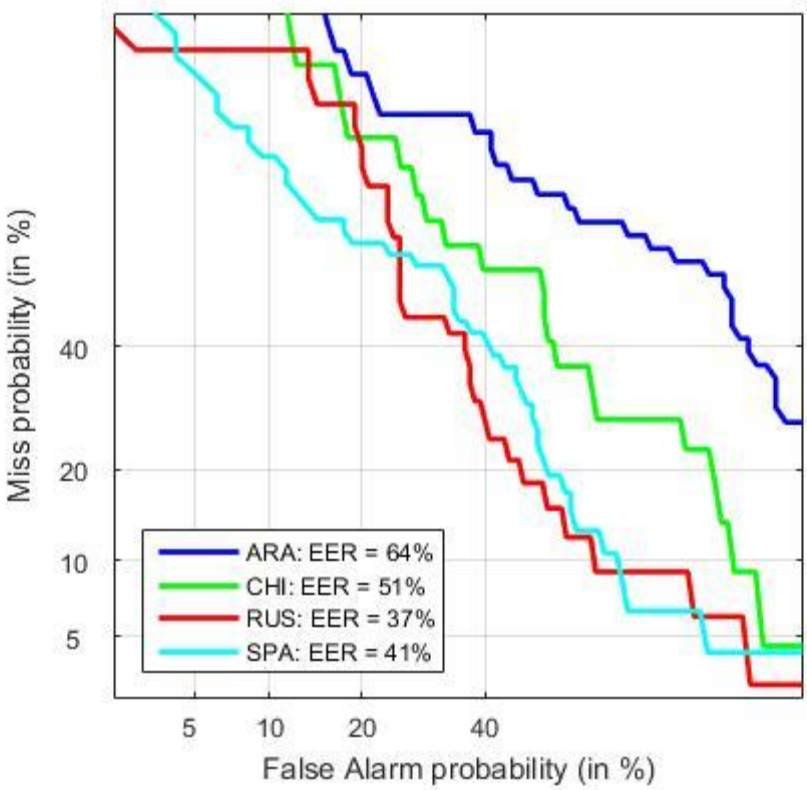

Figura 5.2.3: Curvas DET del sistema de referencia sin mejoras sobre los datos de desarrollo. EER medio: 48.25%

Como podemos ver en la figura 5.2.3, esta primera aproximación está lejos de tener un buen rendimiento debido al EER medio (y de cada acento) tan alto. Podemos decir que las grabaciones tienden a identificarse mejor con el acento ruso y peor con el árabe, aunque hasta este punto no podemos sacar ninguna conclusión y debemos introducir algunas mejoras para mejorar estos resultados.

5.3. Introduciendo mejoras en el sistema de referencia

Como hemos visto en los resultados anteriores, el sistema de referencia creado de esa manera tiene un rendimiento bastante pobre con un EER medio que roza el 50%.
Por ello es necesario introducir mejoras para que ese EER disminuya significativamente. En este trabajo vamos a centrarnos en introducir 4 mejoras: normalización de longitud sobre los *i-vectors*, blanqueamiento, normalización WCCN, y normalización sobre los *scores* finales. A continuación vemos cada una de ellas.

5.3.1. Normalización de longitud (*l-norm)*

Esta normalización es la más sencilla y se puede aplicar directamente sobre todos los *i-vectors* con los que vamos a trabajar; consiste en dividir cada vector por su norma, dada por:

$$\left|\left|a\right|\right| = \sqrt{aa} = \sqrt{\sum_{i=1}^{n} a_i^2}$$

Donde '*a'* es el propio *i-vector*.
Y luego se divide cada elemento del *i-vector* por $\left|\left|a\right|\right|$:

$$i - vector_{final} = \frac{a_i}{\left|\left|a\right|\right|}$$

El porcentaje del EER medio disminuye 0.35 puntos con esta mejora, por lo tanto es casi inapreciable en las curvas DET.
El EER medio tras la normalización de longitud es de 47.9%, excesivamente alto aun.

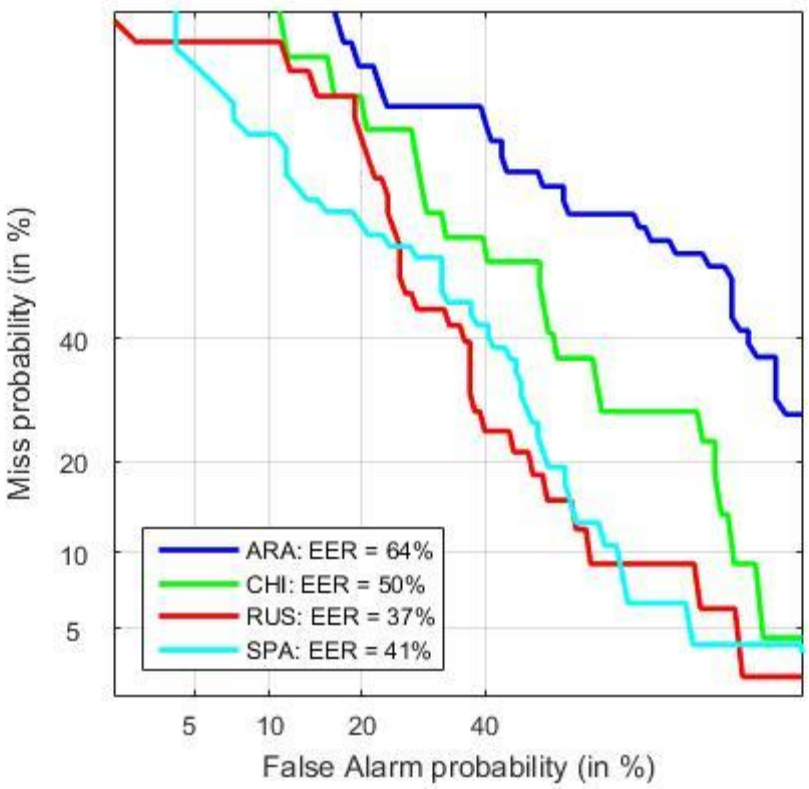

Figura 5.3.1: Curvas DET del sistema de referencia sobre los datos de desarrollo tras normalización de longitud (*l-norm*). EER medio: 47.9%

5.3.2 Blanqueamiento (*whitening*)

Se corresponde con una normalización de media y varianza, pero esa media y esa varianza deben calcularse sobre un conjunto de datos independiente. Calculamos la media y la varianza sobre los *i-vectors* con los que entrenamos los modelos de acento y luego se aplica sobre los *i-vectors* de *test.*

Es la mejora más significativa en cuanto a mejora del rendimiento del sistema se refiere: con ella, logramos disminuir el EER medio final al 24.98%, disminuyendo en 23 puntos el porcentaje de error anterior.

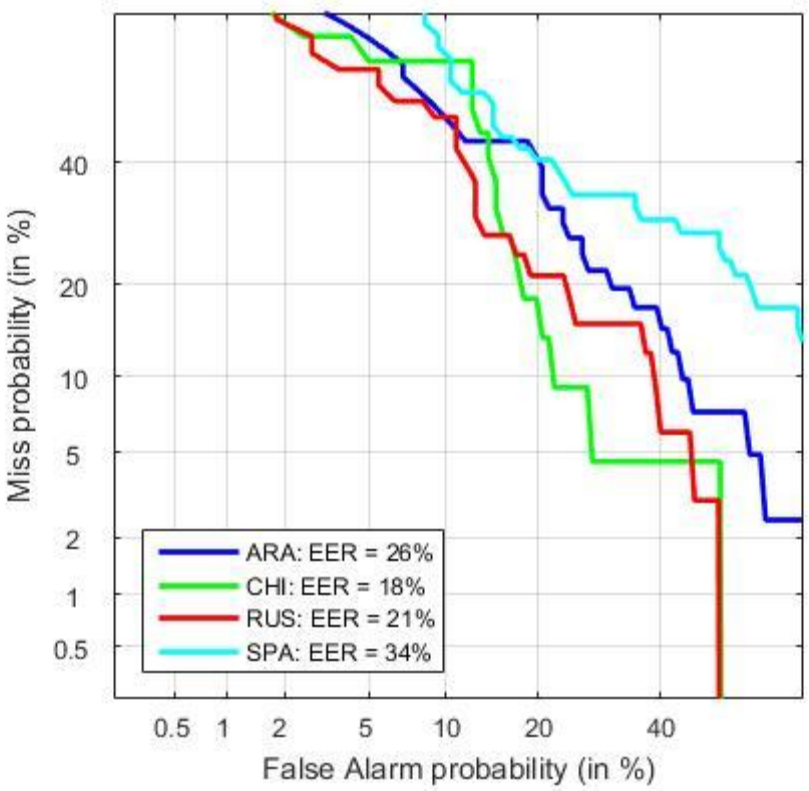

Figura 5.3.2: Curvas DET del sistema de referencia sobre los datos de desarrollo tras blanqueamiento. EER medio: 24.98%

5.3.3 Normalización WCCN

WCCN (*within-class covariance normalization)* se utiliza principalmente para compensar las variaciones dentro de las clases en el espacio de variabilidad total (TV). De la misma manera, se aplica a todo el conjunto de *i-vectors* con los que se trabajan.
La manera de aplicar WCCN es la siguiente; en primer lugar se calcula la matriz WCCN de la siguiente manera:

$$\Lambda = \frac{1}{L}\sum_{a=1}^{L}\frac{1}{N_a}\sum_{i=1}^{N_a}(w_i^a - \overline{w}_a)(w_i^a - \overline{w}_a)^T$$

Donde $\overline{w}_a$ es el *i-vector* media de cada acento 'a', L es el número de acentos, y N_a el número de *i-vectors* del acento 'a'.
La matriz Λ representa la covarianza intra-acento promediada entre los distintos acentos, y se usa para normalizar la variabilidad en las distintas dimensiones del subespacio de variabilidad total (dimensiones de los *i-vectors).* Su inversa puede descomponerse por Cholesky como:
$\Lambda^{-1} = BB^T$.

Siendo esta matriz de transformación 'B' la que se aplica sobre los *i-vectors* en el *kernel* de la distancia coseno para obtener la normalización WCCN final (de forma análoga a como se aplica la proyección LDA).

En este trabajo, la normalización WCCN contribuye a disminuir en 3 puntos el porcentaje de el EER medio que teníamos hasta ahora, consiguiéndolo disminuir hasta el 21.26%.

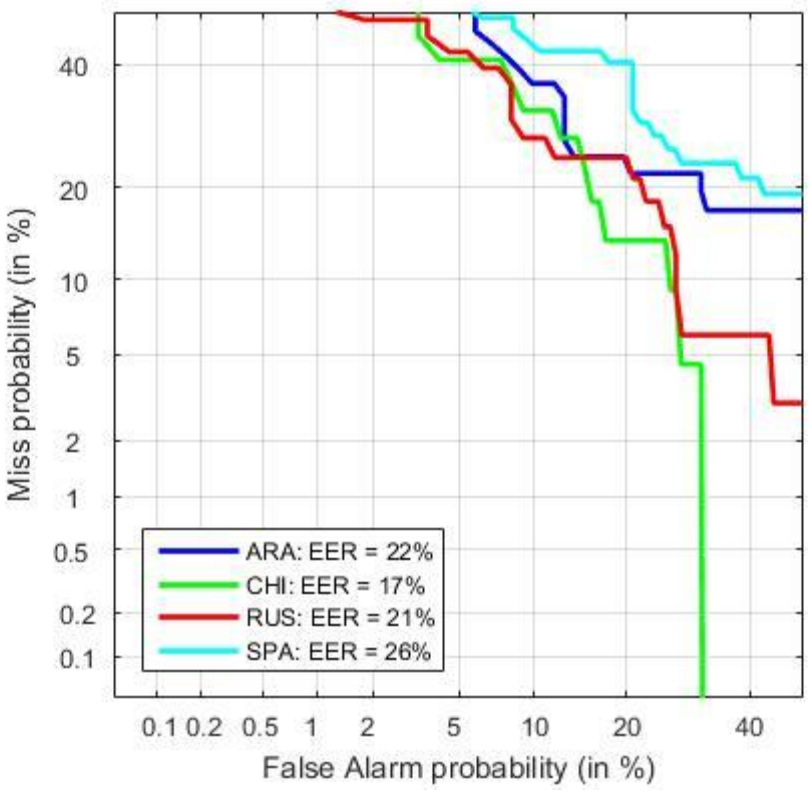

Figura 5.3.3: Curvas DET del sistema de referencia sobre los datos de desarrollo tras normalización WCCN. EER medio: 21.26%

5.3.4 Normalización sobre los *scores* finales

La última mejora que se introduce en este trabajo, a diferencia de las mencionadas anteriormente, se aplica directamente a los *scores* finales y no sobre los *i-vectors.*
Una vez obtenidos los *scores* $\{t_a, a = 1, \dots \dots L\}$ entre una grabación de *test* determinada y los L modelos de acento, la puntuación respecto al acento 'a', se normaliza de la siguiente manera:

$$t'(a) = log \frac{\exp(t_a)}{\frac{1}{L-1}\sum_{k \neq a} \exp(t_k)}$$

Donde t'(a) son los *scores* finales normalizados con los que se van a evaluar el rendimiento del sistema.

En este trabajo, esta última mejora disminuye el porcentaje del EER medio que teníamos hasta ahora en 1 punto, teniendo un EER medio final para el sistema de referencia de 20.12%, habiéndolo conseguido reducir en prácticamente 30 puntos desde el inicio.

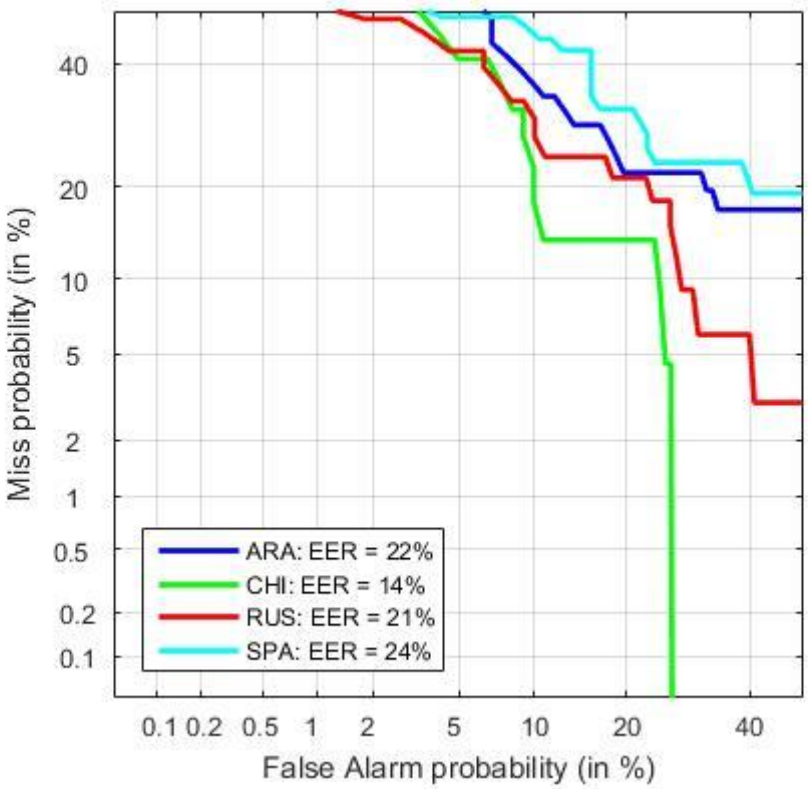

Figura 5.3.4.1: Curvas DET del sistema de referencia sobre los datos de desarrollo tras normalización de los *scores* finales. EER medio: 20.12%

Como se puede ver en la gráfica anterior, unos acentos se detectan mejor (el chino mandarín el que mejor y el español el que peor) que otros, esto puede deberse a que en general los hablantes de un idioma particular tengan más dificultades para hablar inglés, o que en promedio los hablantes de un idioma tienen mejor/peor nivel.

A continuación se muestran los resultados finales del sistema de referencia obtenidos sobre los datos de validación.

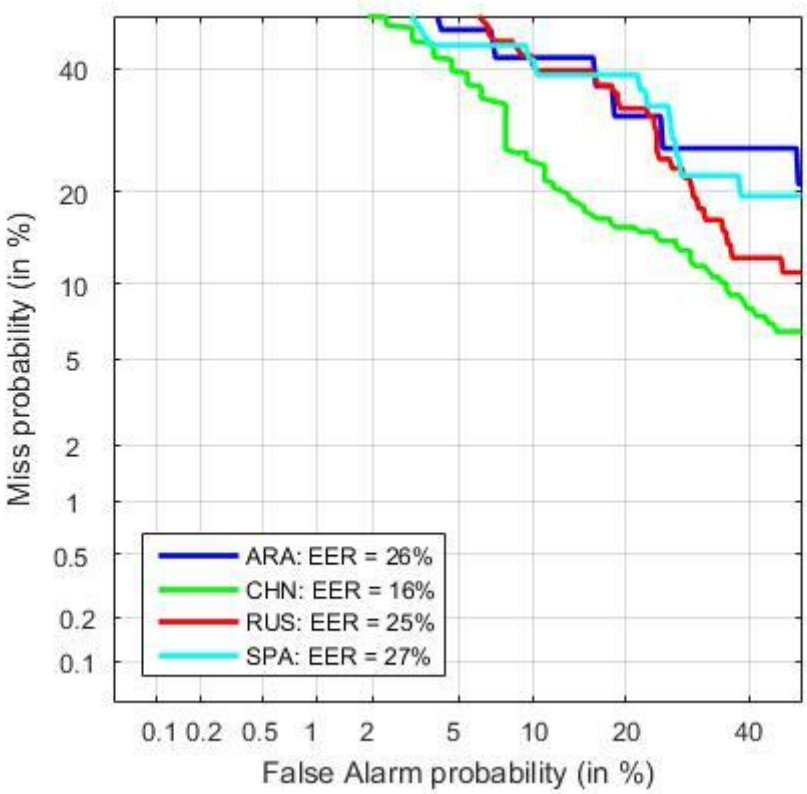

Figura 5.3.4.2: Curvas DET del sistema de referencia sobre los datos de validación tras normalización de los *scores* finales. EER medio: 23.66%

Por último se muestra un resumen de la progresión de los resultados del sistema de referencia vistos hasta ahora con cada una de las mejoras aplicadas

Tabla 5.3.4: Tabla resumen de progresión de resultados para el sistema de referencia.

	+Dist.cos	**+Dist.cos +l-norm**	**+Dist.cos +l-norm +whitening**	**+Dist.cos +l-norm +whitening +WCCN**	**+Dist.cos +l-norm +whitening +WCCN +Normalización *scores***
EERmed desarrollo(%)	48.25	47.9	24.98	21.26	20.12
EERmed validación(%)	48.88	48.52	26.36	25.14	23.66

5.3.5 Comparativa con LDA

Como hemos mencionado en el capítulo 3.2.1, el uso de LDA no influye en una mejora adicional del sistema para este trabajo, por lo que finalmente no se empleó en el mismo.

A continuación se muestran los resultados con todas las normalizaciones y mejoras anteriores más LDA para poner de manifiesto lo mencionado.

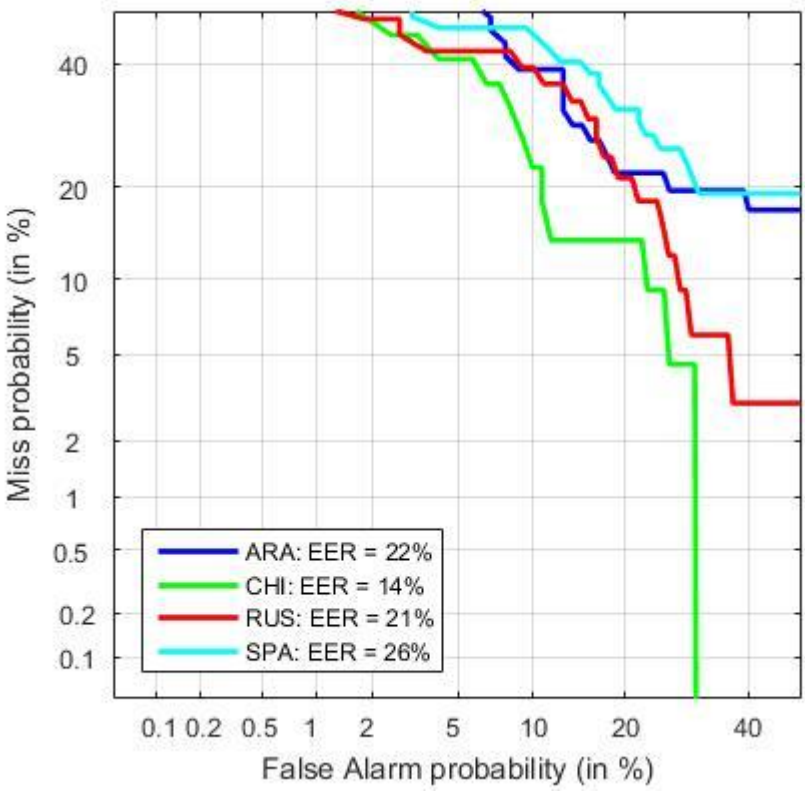

Figura 5.3.5.1: Curvas DET del sistema de referencia sobre los datos de desarrollo aplicando LDA. EER medio: 20.64%

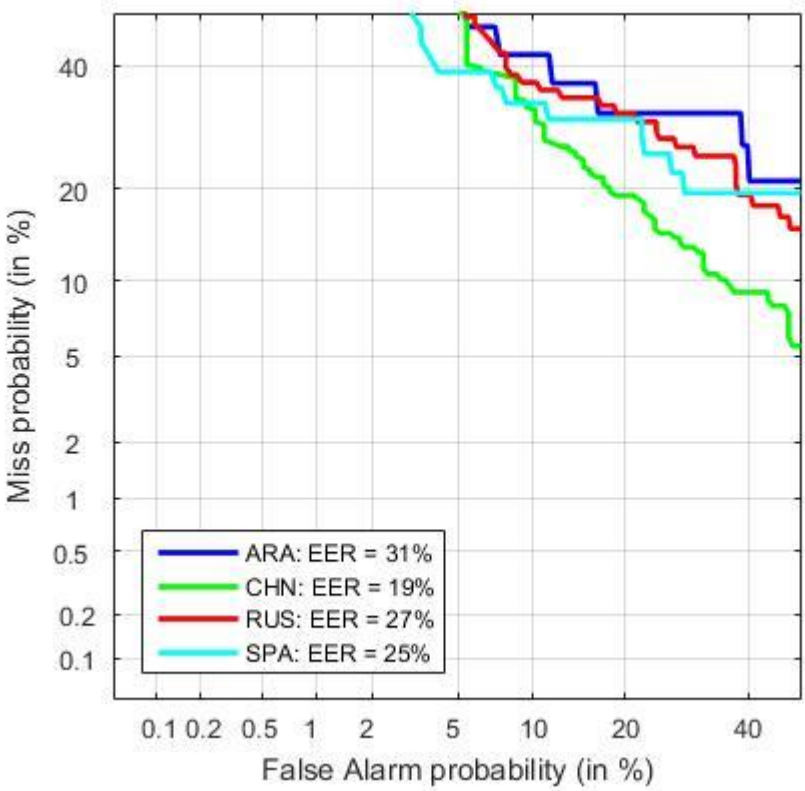

Figura 5.3.5.2: Curvas DET del sistema de referencia sobre los datos de validación aplicando LDA. EER medio: 25.62%

Como podemos ver, el sistema empeora tanto para los datos de desarrollo como para los de validación, como hemos dicho anteriormente, esto puede ser debido a que LDA se basa en disminuir dimensiones a número de clases-1, y al tener sólo 4 acentos la dimensión resultante es muy baja.
En la siguiente tabla se muestra a modo de resumen el rendimiento del sistema con/sin LDA:

Tabla 5.3.5: Comparativa del rendimiento del sistema con/sin LDA.

SISTEMA DE REFERENCIA	**EER SIN LDA (%)**	**EER CON LDA (%)**
NIST-SRE2005 (DESARROLLO)	20.12	20.64
NIST-SRE2006 (VALIDACIÓN)	23.66	25.62

5.4. Creación del sistema basado en unidades lingüísticas

Como hemos dicho en el capítulo 3.3, este sistema sigue el mismo esquema que el sistema de referencia, utilizándolo como punto de partida.

En esta aproximación se aplican las mismas técnicas de normalización y compensación de variabilidad explicadas en los apartados anteriores a cada sistema asociado a una unidad lingüística. Debe tenerse en cuenta que, en esta aproximación, se dispone de un sistema por cada unidad lingüística, proporcionando cada uno de ellos una puntuación para una grabación de test dada. Si bien estos sistemas podrían utilizarse de forma individual (por ejemplo, tomar siempre la puntuación dada por la unidad que mejor funcione para cada acento), la ventaja potencial de esta aproximación reside en poder combinar las puntuaciones de las distintas unidades en una única puntuación final por grabación, lo que se conoce como "fusión de puntuaciones" y generalmente da lugar a un rendimiento mejor que el de cualquiera de los sistemas individuales.

5.4.1. Creación de los modelos de acentos

La creación de los modelos de acento se realiza de la misma forma que en el sistema de referencia pero teniendo en cuenta algunas consideraciones:

El modelo se crea para cada unidad lingüística (por lo tanto se crean hasta 139 modelos), y los *i-vectors* correspondientes a cada unidad tienen dimensiones diferentes (entre 5 y 50). Por lo tanto el modelo ya no es uno único de dimensiones 600x4 si no que ahora varía dependiendo la unidad entre 5-50 x 4 (correspondientes a los 4 acentos que no cambiamos).
También tenemos que tener en cuenta que no todas las grabaciones de la base de datos correspondientes a los ficheros de *train* con los que se entrenan los modelos tienen todas las unidades lingüísticas. Esto quiere decir que el modelo no se va a entrenar con los 1951 *i-vectors* como se hacía anteriormente, si no que dependiendo de si la unidad está o no presente en la grabación, el modelo se entrenará con más o menos datos.

5.4.2. *Scores* entre el modelo y los datos de *test*

Hay que tener en cuenta que para los ficheros de *test* ocurre lo mismo que para los de *train:* no todas las unidades lingüísticas van a estar presentes en ellos.
Como luego queremos combinar las puntuaciones de distintos sistemas, es necesario tener una puntuación para cada uno de ellos (para cada unidad lingüística); nosotros en este trabajo resolvemos este problema poniendo a cero la puntuación para la unidad que no aparece en la grabación de test, que es una primera aproximación al problema (en el capítulo 6.2 se cuenta otra posible aproximación), esto lo hacemos así por problemas de concordancia de dimensiones más adelante a la hora de hace la fusión de unidades.

5.4.3. Resultados antes de la fusión

Antes de hacer la fusión de unidades, obtenemos resultados correspondientes a las 139 unidades lingüísticas por separado. Estos resultados obtenidos sobre la partición de desarrollo suelen presentar un EER medio muy alto por lo general (ver anexo). Esto es debido a que una unidad lingüística puede ser muy discriminativa para un acento pero muy poco para los otros, y en promedio acabamos teniendo EERs altos. Con ello, podemos observar es más eficaz reconocer un acento a partir de una o varias unidades lingüísticas

específicas, y ver que son distintas para cada acento. A continuación se muestra una tabla donde se muestran los EERs de las 5 mejores unidades, ordenadas de mejor (menor EER) a peor para cada acento.

Tabla 5.4.3: Unidades con menor EER para cada acento sobre los datos de desarrollo.

ARABE		**CHINO MANDARIÍN**		**RUSO**		**ESPAÑOL**	
Unidad	***EER (%)***	***Unidad***	***EER (%)***	***Unidad***	***EER (%)***	***Unidad***	***EER (%)***
L	25.3	WAH	17.2	L	28.0	R	24.0
R	25.4	TUW	22.1	AY	31.4	RIY	28.8
RIY	29.0	P	22.1	AYT	35.1	AWT	31.1
UW	31.1	L	22.3	LAY	35.3	ER	32.5
AXL	31.7	AYT	22.4	N	35.6	L	32.8

A continuación se muestran las curvas DET para las 5 unidades lingüísticas con menor EER en promedio:

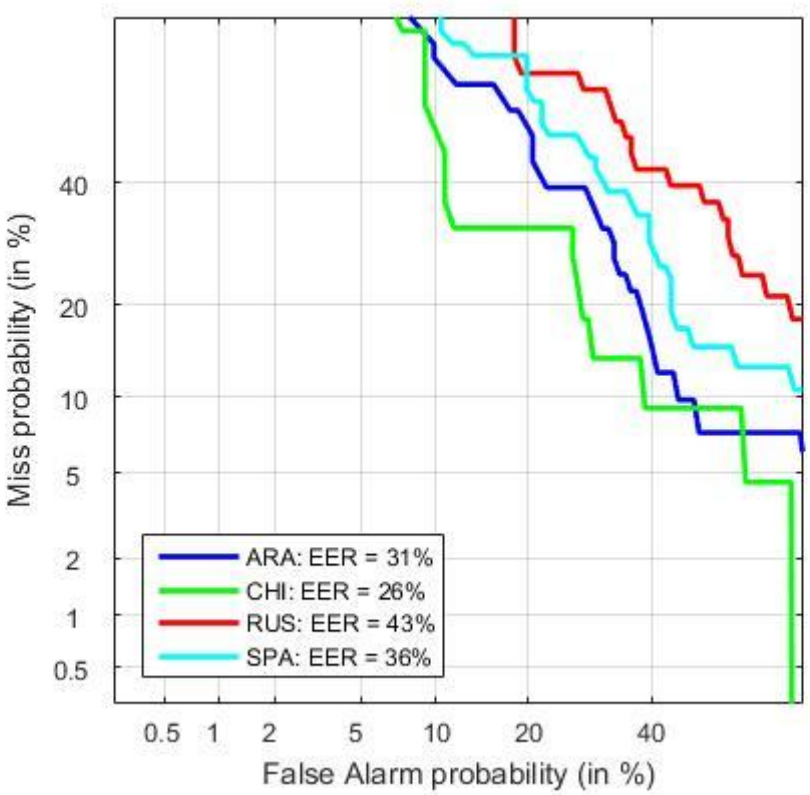

Figura 5.4.3.1: Curvas DET del sistema para la unidad lingüística 'AXL'. EER medio: 34.25%

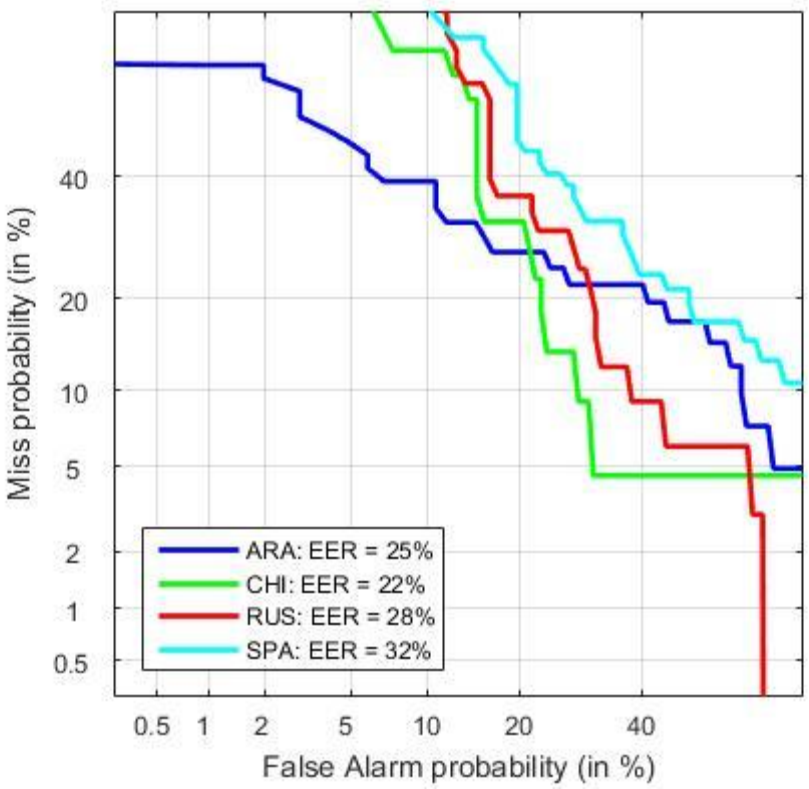

Figura 5.4.3.2: Curvas DET del sistema para la unidad lingüística 'L'. EER medio: 26.82%

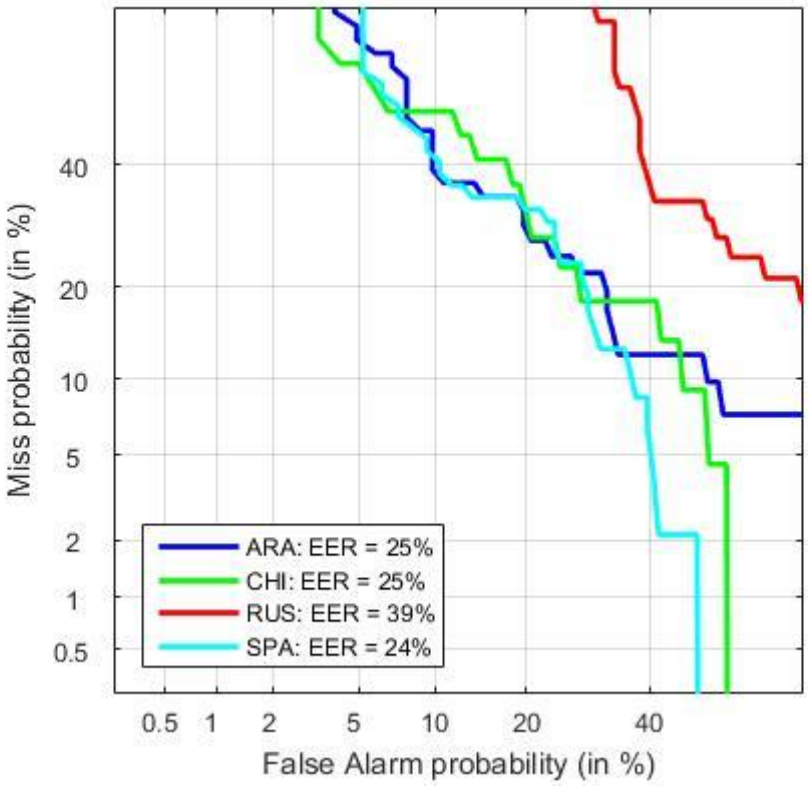

Figura 5.4.3.3: Curvas DET del sistema para la unidad lingüística 'R'. EER medio: 28.08%

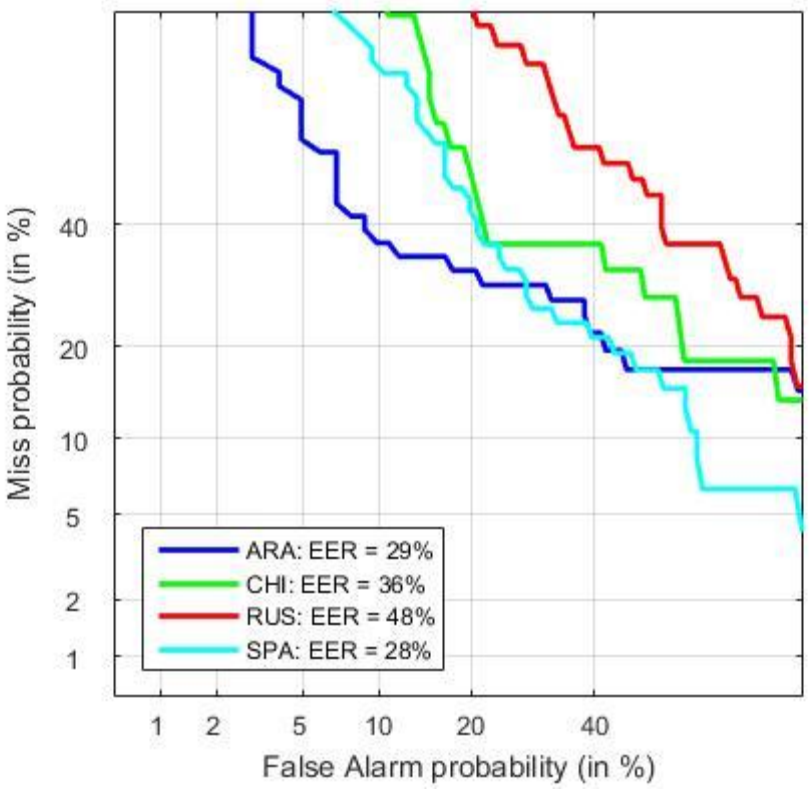

Figura 5.4.3.4: Curvas DET del sistema para la unidad lingüística 'RIY'. EER medio: 35.52%

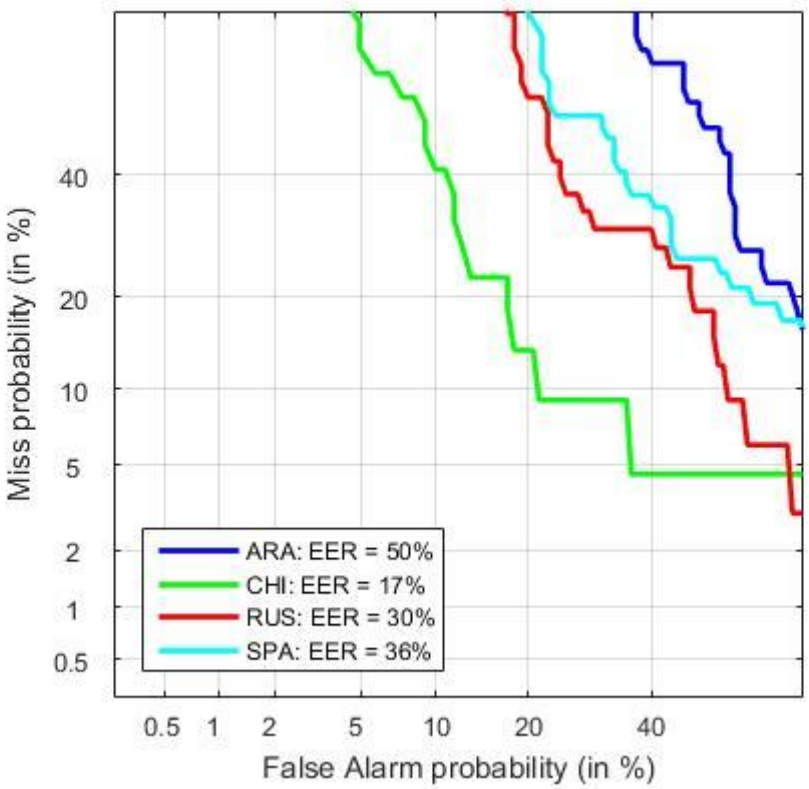

Figura 5.4.3.5: Curvas DET del sistema para la unidad lingüística 'WAH'. EER medio: 33.45%

5.4.4. Fusión de unidades lingüísticas

La fusión de unidades consiste en la combinación de puntuaciones dadas por cada sistema independiente, asociado cada uno a una unidad lingüística.
En este trabajo se entrena una única fusión, así que el resultado es el mejor resultado "promedio" sobre todos los acentos (en el capítulo 6.2 veremos otra forma de entrenar la fusión).

Para llevar a cabo la fusión, en primer lugar debemos extraer todas las puntuaciones *target (scores* del fichero de *test* que corresponden con el acento real del mismo, es decir, 1 *score* por fichero) y *non-target* (*scores* del fichero de *test* que no se corresponden con el acento real del mismo, es decir, 3 *scores* por fichero) de todas las unidades lingüísticas con las que trabajamos, con el fin de entrenar dicha fusión por regresión logística. Estas puntuaciones nos van a permitir calcular los coeficientes óptimos de la fusión (w).
Es importante recordar que estos coeficientes o pesos (w) se calculan a partir de los *scores* obtenidos en NIST-SRE2005, pero para obtener la fusión final de unidades, aplicamos dichos coeficientes sobre los *scores* del conjunto de *i-vectors* pertenecientes a NIST-SRE2006 (la fusión se obtiene multiplicando los coeficientes 'w' por las puntuaciones).

5.4.5. Resultados después de la fusión

Finalmente, tras obtener la fusión de unidades lingüísticas, evaluamos los resultados: la fusión mejora prácticamente un 3% el mejor resultado que teníamos hasta ahora en el sistema de referencia, llegando a reducir el EER medio hasta el 17.51% y alcanzando de esta forma el mejor resultado final del trabajo.

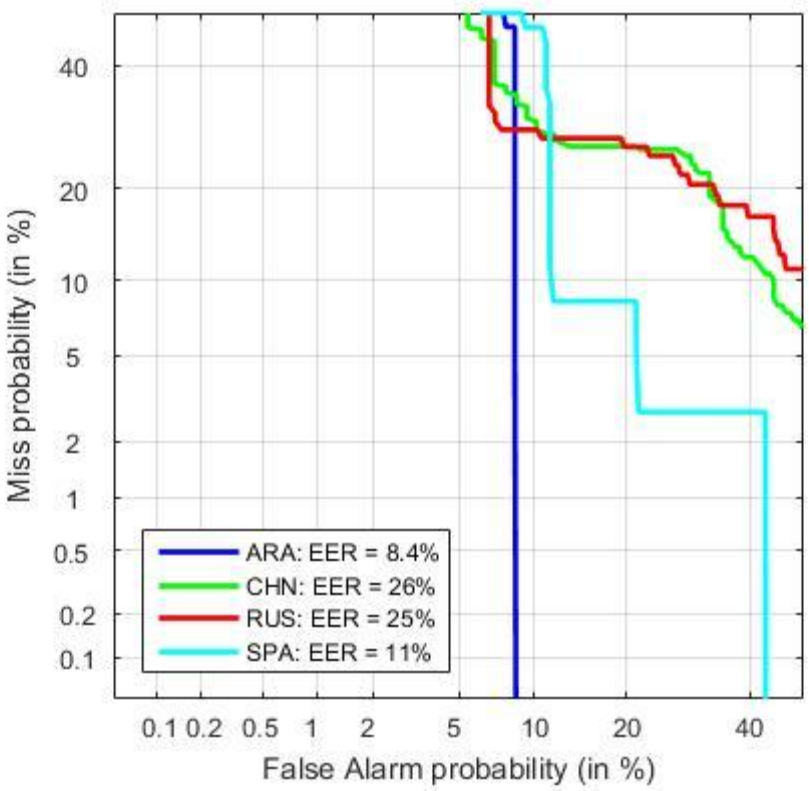

Figura 5.4.5: curvas DET finales tras aplicar la fusión de unidades lingüísticas. EER medio: 17.51%

6 Conclusiones y trabajo futuro

6.1 Conclusiones

En base a los resultados obtenidos, podemos afirmar que la aproximación basada en la fusión de puntuaciones procedentes de sistemas individuales que modelan las frecuencias formantes en unidades lingüísticas mejora sensiblemente al sistema de referencia, el cual modela la información acústica en todo el fichero de forma global, a cambio de un mayor coste computacional en función de la cantidad de unidades lingüísticas bajo análisis.

En reconocimiento de acento, los resultados y el rendimiento de los sistemas siempre van a depender del nivel de los locutores con el idioma con el que fueron entrenados (en este trabajo inglés). Un mayor nivel del idioma puede llevar a una falsa identificación del acento, por ello es necesario trabajar sobre bases de datos más adecuadas a este propósito, donde los hablantes de una lengua materna determinada tengan claramente un marcado acento que sea "reconocible por humanos", o al menos se proporcione un etiquetado manual que indique el "grado de acento" de un hablante (que sería, por ejemplo, 0 en el caso de un hablante bilingüe, sin acento) para posteriormente crear sistemas más robustos y que induzcan menos a error. En el apartado 6.2 introducimos algunas pistas sobre este aspecto.

La información lingüística que explota la aproximación usada permite identificar mejor el acento de un hablante para unidades lingüísticas determinadas, ya que dependiendo de la lengua materna se diferencian más unos fonemas o difonemas que otros como hemos visto en los resultados del capítulo 5.4.
Viendo los resultados del sistema basado en unidades lingüísticas, con sólo una unidad, específica para cada idioma, podemos tener tan buen rendimiento como el sistema de referencia. Este hecho se puede ver, por ejemplo, con el fonema "r" dónde se ve que es más discriminativo para el español (figura 5.4.3.3) porque los hablantes españoles pueden tender a hacerlo "más fuerte" (de hecho en el español lo es), y lo mismo ocurre, por ejemplo, para el fonema "wah" (figura 5.4.3.5) con el acento chino.

También podemos afirmar que el rendimiento del sistema está sujeto al conjunto de normalizaciones que se aplican sobre él, tanto para los datos como para las puntuaciones finales. Este conjunto de normalizaciones se traducen en mejora, y podemos destacar el blanqueamiento como la que introduce un mayor salto de calidad en el sistema.

6.2 Trabajo futuro

En este trabajo hemos visto diversas mejoras que ayudan a aumentar el rendimiento del sistema disminuyendo cada una el EER medio tanto independientemente como conjuntamente. A continuación veremos algunas otras mejoras que se podrían introducir para seguir mejorando el sistema en algún trabajo futuro.

Como hemos mencionado anteriormente, en este trabajo la aplicación de LDA (*linear discriminant analysis)* no mejora los resultados, posiblemente debido a la baja dimensionalidad de los vectores resultantes al tener sólo 4 acentos. Posiblemente, si se

trabajase con un entorno experimental con más acentos involucrados, LDA (que busca disminuir la dimensionalidad), funcionaría de manera más adecuada.
Además, existe una mejora a LDA denominada HDLA (*heteroscedasticlinear discriminant analysis)* [2] que también se basa en reducir la dimensionalidad de los *i-vectors*, y que a diferencia de LDA trata con la información discriminante presentes en las matrices de medias y covarianzas, y que si se utilizase podría obtener resultados más favorables.

Por otro lado, en este trabajo hemos considerado que las distribuciones de scores target y non-target se cruzan en un valor de score igual a 0 (*Figura 6.2*), y por ello, cuando una unidad lingüística no existía para una grabación dada, sustituíamos el score por dicho valor, asociándolo a una decisión "neutra" sobre la comparación respecto a un acento dado. Esto no tiene porqué ser así, y el procedimiento adecuado sería analizar dichas distribuciones para determinar ese valor de score o, siendo más precisos, convertir dichas puntuaciones a relaciones de verosimilitud (LRs) mediante un proceso de calibración [9] y asociar a dichas comparaciones el valor LR = 1, que equivale a no apoyar ninguna de las hipótesis alternativas (la grabación de test corresponde al acento, o no corresponde al acento).). Esta aproximación estaría menos sujeta a un sesgo en la elección del *score* "neutro", que puede estar influyendo en el entrenamiento de la fusión de puntuaciones.

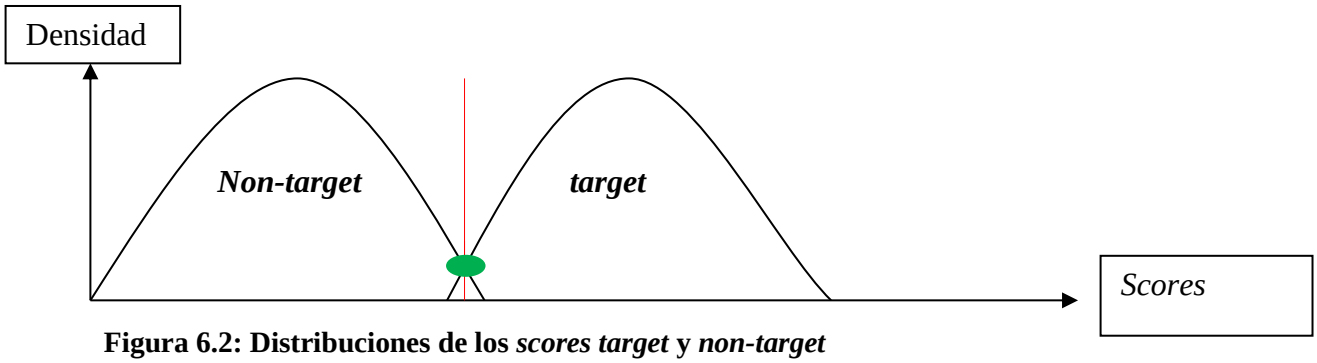

Figura 6.2: Distribuciones de los *scores target* y *non-target*

También, como hemos comentado en el apartado 5.4.5, en este trabajo se entrena una única fusión, con lo que el resultado es el mejor resultado promedio sobre todos los acentos, pero podría ser mucho mejor entrenar una fusión para cada acento, ya que hemos visto que el rendimiento de cada unidad varía mucho de un acento a otro.

Por último, podría ser de gran interés implementar un protocolo experimental que involucre más comparaciones por acento que el actual, por ejemplo a través del método LOSO (*Leave one Speaker Out).*

Referencias

[1] Mohamad Hasan, Rahim Saeidi, Hugo Van hamme, David Van Leeuwen, "Accent recognition using i-vector, Gaussian mean supervector and Gaussian posterior probability supervector for spontaneous telephone speech", Center for processing speech and images,KU Leuven, Belgium.

[2] Hamid Behravan, Ville Hautamaki, Tomi Kinnunen, "Factors affecting i-vector based foreign accent recognition: A case study in spoken Finnish". School of computing, University of Eastern Finland, October 2014.

[3] M.Li, K.J Han, and S.Narayanan, "Automatic speaker age and gender recognition using acoustic and prosodic level information fusion". Computer Speech and Language, vol 27, no 1, pp151-167, 2013.

[4] N.Dehak, P.A Torres-Carrasquillo, D.Reynolds, and R.Dehak, "Language recognition via ivectors and dimensionality reduction". In Proc.Interspeech, 2011, pp. 857-860.

[5] Najim Dehak and Stephen Shum "Low-dimensional speech representation based on Factor Analysis and its applications". Spoken Language System Group. MIT Computer Science and Artificial Intelligence Laboratory.

[6] A.Hanani, "Human and computer recognition of regional accents and ethnic groups from british English speech" University of Birmingham, July 2012.

[7] F.Biadsy, "Automatic dialect and accent recognition and its application to speech recognition". Columbia University, 2011.

[8] Georgina Brown "Automatic accent recognition systems and the effects of data on performance". Department of Language and Linguistic science, University of York, UK.

[9] Javier Franco-Pedroso, Joaquín Gonzalez-Rodriguez, "Linguistically-constrained formant-based i-vectors for automatic speaker recognition". ATVS Biometric Recognition Group, EPS UAM, November 2015.

[10] Kajarekar,S.S.,Scheffer,N.,Graciarena,M.,Shriberg,E.,Stolcke,A.,Ferrer,L.,Bocklet,T. ,2009. The SRI NIST 2008 speaker recognition evaluation system. In: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2009), 19-24April2009, Taipei, Taiwan, pp. 4205–4208.

[11] Goldwater,S.,Jurafsky, D,Manning, C.D. ,2010. Which words are hard to recognize? Prosodic, lexical, and disfluency factors that increase speech recognition error rates. Speech Commun.52(3),181–200.

Glosario

EER	*Equal Error Rate*
DET	*Detection Error Tradeoff*
LDA	*Linear Discriminant Analysis*
TV	*Total Variability*
WCCN	*Within-class Covariance Normalization*
SRE	*Speaker Recognition Evaluation*
HLDA	*Heteroscedastic linear Discriminant Analysis*

Anexos

A Unidades lingüísticas y EER para cada acento

	EER (%)			
Unidad	**ÁRABE**	**CHINO M.**	**RUSO**	**ESPAÑOL**
AA	49	39	49	41
AAR	40	41	45	43
AAT	48	41	40	46
AE	42	27	39	49
AEN	45	60	46	45
AET	46	43	40	47
AEV	48	50	52	56
AH	62	26	61	74
AHM	50	34	54	61
AHN	52	36	61	57
AHT	43	34	48	53
AO	46	41	49	42
AOL	43	41	43	51
AOR	33	30	52	41
AW	54	30	44	41
AWT	45	36	45	31
AX	43	41	41	45
AXB	50	27	40	56
AXD	54	36	45	64
AXG	51	45	65	53
AXK	47	41	47	61
AXL	31	26	43	36
AXM	54	45	53	59
AXN	54	48	55	56
AXNG	50	53	45	55
AXS	52	50	45	43
AXT	57	39	39	65
AXV	46	43	41	51
AXZ	56	40	38	63
AY	53	32	31	55
AYD	51	47	44	48
AYK	69	35	43	59
AYM	55	30	51	57
AYN	48	54	48	59
AYT	64	22	35	46
B	50	26	48	61
BAH	46	41	48	58
BAX	51	27	61	58
BIY	44	41	48	50

CH	50	50	52	52
D	36	33	44	57
DAX	56	41	44	55
DDH	46	44	50	47
DH	46	31	39	61
DHAE	50	41	45	49
DHAX	53	41	48	58
DHEH	39	41	48	43
DHEY	53	36	48	66
DIH	61	45	38	57
DOW	47	45	54	50
DUW	52	36	48	59
DX	47	31	43	47
DXAX	54	41	43	49
DXIY	52	44	43	46
EH	34	29	45	40
EHL	38	41	52	39
EHN	43	30	49	57
EHR	40	45	49	34
ER	35	32	52	32
EY	52	31	50	57
F	54	32	43	47
G	45	25	57	57
HH	52	39	36	59
HHAE	54	48	44	47
HHW	54	36	48	46
IH	44	36	46	53
IHN	49	40	64	60
IHNG	47	45	39	67
IHT	54	41	45	54
IY	51	30	44	39
IYAX	46	39	52	57
IYN	45	32	38	44
IYP	64	41	46	55
JH	47	32	45	48
JHAX	51	45	57	64
K	43	35	48	60
KAH	54	32	51	49
KAX	58	45	44	45
KS	50	60	55	53
L	25	22	28	32
LAX	44	40	42	48
LAY	44	45	35	55
LIY	50	41	52	50
M	48	27	36	47
MAX	64	45	53	55
MAY	56	32	58	59
MIY	47	30	39	50

N	51	36	35	43
NAA	40	41	55	45
NAX	53	43	56	58
ND	46	45	47	57
NDH	49	60	45	63
NG	42	36	42	60
NGK	61	40	52	64
NIY	46	49	39	49
NOW	45	45	40	57
NS	49	59	52	47
NT	53	45	48	59
OW	49	41	45	52
OWN	51	50	57	61
P	42	22	39	46
PAX	47	45	48	45
PUH	43	60	56	47
PUM	44	34	45	55
R	25	25	39	24
RAX	34	46	63	35
RAY	49	25	47	51
RIY	29	36	48	28
S	47	43	45	47
SAH	51	41	49	42
SAX	54	41	35	48
SH	54	56	43	42
SOW	42	29	42	36
ST	44	48	47	34
T	34	40	45	38
TAX	45	32	45	42
TAY	54	32	39	48
TDH	51	49	48	56
TH	53	45	52	59
THIH	52	43	48	69
TR	46	50	52	45
TS	38	49	52	51
TUW	44	22	44	59
TW	50	43	52	43
UH	55	52	55	39
UHD	39	48	53	44
UW	31	41	48	39
UWN	55	47	55	68
V	50	36	52	59
VAX	54	41	45	52
W	43	29	38	42
WAH	50	17	30	36
WAX	49	37	55	53
WEH	37	33	55	38
Y	51	41	47	60

YAE	48	28	52	49
YUW	39	30	48	46
Z	40	41	39	45
ZAX	45	43	50	47